ELSEVIER

Theoretical Computer Science 173 (1997) 311–347

Theoretical Computer Science

May I borrow your logic? (Transporting logical structures along maps)

Maura Cerioli [a,*], José Meseguer [b,1]

[a] *DISI-Dipartimento di Informatica e Scienze dell'Informazione, Università di Genova, Via Dodecaneso, 35, I-16146 Genova, Italy*

[b] *SRI International, Menlo Park, CA 94025, and Center for the Study of Language and Information, Stanford University, Stanford, CA 94305, USA*

Abstract

It can be very advantageous to borrow key components of a logic for use in another logic. The advantages are both conceptual and practical; due to the existence of software systems supporting mechanized reasoning in a given logic, it may be possible to reuse a system developed for one logic – for example, a theorem-prover – to obtain a new system for another. Translations between logics by appropriate mappings provide a first natural way of reusing tools of one logic in another. This paper generalizes this idea to the case where entire components – for example, the proof theory – of one of the logics involved may be completely missing, so that the appropriate mapping could not even be defined. The idea then is to borrow the missing components (as well as their associated tools if they exist) from a logic that has them in order to create the full-fledged logic and tools that we desire. The relevant structure is transported using maps that only involve a limited aspect of the two logics in question – for example, their model theory. The constructions accomplishing this kind of borrowing of logical structure are very general and simple. They only depend upon a few abstract properties that hold under very general conditions given a pair of categories linked by adjoint functors.

1. Introduction

The use of logic in computer science is undergoing vigorous growth. Since the applications are many, there are increasingly stronger interactions between the two fields that are having a profound impact on both of them. New logics are frequently

* Corresponding author. E-mail: cerioli@disi.unige.it. Partially supported by Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo of CNR (Italy), MURST – 40% Modelli e Specifiche di Sistemi Concorrenti and by HCM-Medicis.

[1] Partially supported by the US Office of Naval Research under contracts N00014-92-C-0518 and N00014-94-1-0857, NSF Grant CCR-9224005, and by the Information Technology Promotion Agency, Japan, as a part of the Industrial Science and Technology Frontier Program "New Models for Software Architecture" sponsored by NEDO (New Energy and Industrial Technology Development Organization).

PII S0304-3975(96)00160-0

being proposed, and new variants or adaptations of existing logics for new purposes are widespread.

This proliferation of logics – although certainly a sign of vitality and intellectual creativity – brings with it important conceptual challenges. In a sense, each logic is a different language and, as in the case of natural languages, there is often a serious need to bridge the gap between different languages by means of appropriate translations, and the danger of serious confusion when translations are not correct. There is also a related need to understand the essential features shared by logics in general so that systematic methods can be developed to deal with these problems.

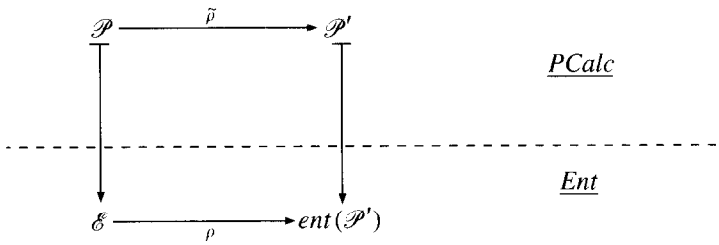
In computer science, the conceptual needs posed by the proliferation of logics were first addressed by Goguen and Burstall [11], who proposed their theory of *institutions* as a general framework for logics. The work on institutions has been further developed by their original proponents and by others [12, 13, 33, 34], and has influenced other notions proposed by different authors [23, 28, 7, 24, 17, 29, 5, 1]. Some of the notions proposed are closely related to institutions; however, in other cases the main intent is to substantially expand the primarily model-theoretic viewpoint provided by institutions to give an adequate treatment of proof-theoretic aspects such as entailment and proof structures.

Institutions arose out of work on the Clear specification language [2], in which the goal was to provide powerful modularity and parameterization mechanisms to structure and reuse formal specifications. Such reusability techniques have later been applied to a good number of specification and logical programming languages such as, for example, [8, 15, 6, 30, 14, 25]. However, the need for reusability arises not only inside one logic – so that specifications or logical programs written in that logic can be reused – but also at the metalevel, in the sense that it can be greatly advantageous to reuse entire logics, or key components of such logics. The advantages may be not only conceptual, although of course this is important; due to the existence of software systems, supporting mechanized reasoning in a given logic, it may be possible to reuse a system developed for one logic – for example, a theorem-prover – to obtain a new system for another.

Translations between logics by appropriate mappings – especially if they are *conservative* in the sense of [24] – provide a first natural way of reusing tools of one logic in another, by translating the appropriate sentences or proofs and using the original tool on the translations. This paper generalizes this idea to the case where entire components – for example, the proof theory – of one of the logics involved may be completely missing, so that the appropriate mapping could not even be defined. The idea then is to borrow the missing components (as well as their associated tools if they exist) from a logic that has them in order to create, ex nihilo as it were, the full-fledged logic and tools that we desire. The relevant structure is transported using maps that only involve a limited aspect of the two logics in question – for example, their model theory.

The constructions accomplishing this kind of borrowing of logical structure are very general and simple. We show that they only depend upon a few abstract properties that hold under very general conditions given a pair of categories linked by adjoint functors. Therefore, the constructions capitalize on the fact that, as was shown in [24], the different components of a logic – entailment relation, model theory, and proof theory – are in a very precise technical sense *modular*, namely in that they can be added or deleted by means of constructions that are adjoint functors.

Consider for example the case where only the consequence relation component of a logic – what we call an entailment system $\mathscr{E}$ – is known, but we have another logic for which a proof theory – what we call a proof calculus $\mathscr{P}'$ – has been fully specified. Since a proof calculus also specifies a logic's consequence relation, there is a forgetful functor $ent : \underline{PCalc} \to \underline{Ent}$ from the category of proof calculi to that of entailment systems. Suppose that at the entailment system level we have a map $\rho : \mathscr{E} \to ent(\mathscr{P}')$. Then, our borrowing construction endows $\mathscr{E}$ with a proof calculus $\mathscr{P}$ borrowed from $\mathscr{P}'$ via ρ such that $ent(\mathscr{P}) = \mathscr{E}$, and ρ lifts to a map $\tilde{\rho}$



such that $ent(\tilde{\rho}) = \rho$. In fact we show that $\tilde{\rho}$ is an *optimal* lifting that satisfies an adequate universal property among all liftings, namely that $\tilde{\rho}$ is a cartesian lifting of ρ, and that *ent* is a *fibration* (for the concept of fibration see [16, 19]). We show that the exact same cartesian lifting property holds not only for the functor *ent*, but also for any of the forgetful functors that disregard some component or components of a logic's structure. There are seven such forgetful functors; we show that for each of them there is an optimal borrowing of the missing logical structure of the kind just described.

Moreover, the forgetful functors discarding some logical components are not only fibrations, but fibrations of a special kind, that are obtained using the fact that they are all left adjoints. We give a general construction, called *extension*, that generalizes adjunctions to comma categories and then obtain the fibration property as a special case.

For each such borrowing construction we give an explicit description of the new logical structure being borrowed and prove that, in general or under natural assumptions,

the new structure thus obtained satisfies particularly good properties such as completeness of the logic, conservativity of the cartesian lifting, or exclusive dependence of the new structure on the one being borrowed. Relevant examples illustrating the usefulness of these constructions are also given.

The methods proposed in this paper should be regarded as a concrete step towards the goal of *formal interoperability* [26], that is, the capacity to move in a mathematically rigorous way across the different formalizations of a system, and to use in a rigorously integrated way the different tools supporting such formalizations.

In particular, the results in this paper have useful applications to the notion of *logical framework*, that is, a logic $\mathscr{F}$ inside which many other logics can be faithfully represented by adequate mappings (see [26] for a discussion of logical frameworks consistent with the ideas in this paper). Indeed, our results show that representations into a logical framework can be defined in a very economic way by mappings preserving a minimum of logical structure – for example, preserving just the consequence relation – since all the remaining logical structure enjoyed by the framework – such as its proof theory or its models – as well as any supporting tools available for $\mathscr{F}$, can then be borrowed in an automatic way.

The paper is organized as follows. Section 2 recalls the basic general notions of logic used in the paper. Section 3 states and proves the general category-theoretic results underlying the desired transportation of logical structure. Section 4 applies the general results to the borrowing of different components. We end with some concluding remarks in Section 5.

2. General logics

Since the significant examples of application of the general categorical construction are all based on the concepts of *institution* [11] and *general logic* [24], this section is devoted to recalling some basic definitions and results from these theories and to proving two new results (Propositions 9 and 10) needed in this paper. More detailed discussions of institutions and general logics can be found in [13, 24].

Institutions cover the semantic aspects of a logical framework, providing formal counterparts for the notions of *signature*, *sentences*, *models* and *validity*, while entailment systems deal with the deductive part. Putting together an institution and a compatible entailment system, i.e. an entailment system that is sound w.r.t. the institution, a *logic* is obtained, where tools to deal with provability and with model-theoretic aspects are both at hand. But, since entailment systems focus only on provability as a consequence relation and abstract away any other proof-theoretic aspects, the concept of (structured) proof is not formalized; thus *proof calculi* are introduced to cover also this feature.

Definition 1. An *entailment system*[2] [24, Definition 1], is a triple $\mathscr{E} = (\mathbf{Sign}, Sen, \vdash)$ consisting of:

- a category **Sign**, whose objects are called signatures;
- a functor $Sen: \mathbf{Sign} \to \mathbf{Set}$, giving the set of *sentences* over a given signature;
- a function $\vdash$ associating to each Σ in **Sign** a binary relation $\vdash_\Sigma \subseteq \wp(Sen(\Sigma)) \times Sen(\Sigma)$, called *$\Sigma$-entailment*, satisfying the following properties:
  - (i) *reflexivity*: for any $\phi \in Sen(\Sigma)$, $\{\phi\} \vdash_\Sigma \phi$;
  - (ii) *monotonicity*: if $\Gamma \vdash_\Sigma \phi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash_\Sigma \phi$;
  - (iii) *transitivity*: if $\Gamma \vdash_\Sigma \phi_i$ for all $i \in I$ and $\Gamma \cup \{\phi_i \mid i \in I\} \vdash_\Sigma \psi$, then $\Gamma \vdash_\Sigma \psi$;
  - (iv) *$\vdash$-translation*: if $\Gamma \vdash_\Sigma \phi$, then $Sen(\sigma)(\Gamma) \vdash_{\Sigma'} Sen(\sigma)(\phi)$ for any $\sigma: \Sigma \to \Sigma'$ in **Sign**.

Definition 2. An *institution* [11, Definition 14], is a 4-tuple $\mathscr{I} = (\mathbf{Sign}, Sen, Mod, \models)$ consisting of:

- a category **Sign** of *signatures*;
- a functor $Sen: \mathbf{Sign} \to \mathbf{Set}$, giving the set of *sentences* over a given signature;
- a functor $Mod: \mathbf{Sign}^{op} \to \mathbf{Cat}$, giving the category of *models* of a given signature;
- for each Σ in **Sign** a *satisfaction* relation $\models_\Sigma \subseteq |Mod(\Sigma)| \times Sen(\Sigma)$, such that, for each morphism $\sigma: \Sigma \to \Sigma'$ in **Sign**, the *Satisfaction Condition*

$$M' \models_{\Sigma'} Sen(\sigma)(\xi) \Leftrightarrow Mod(\sigma)(M') \models_\Sigma \xi$$

holds for each $M' \in |Mod(\Sigma')|$ and each $\xi \in Sen(\Sigma)$.

Given an entailment system (respectively, an institution) its category $\mathbf{Th}_0$ of *theories* has as objects pairs $T = (\Sigma, \Gamma)$ with Σ a signature and Γ a set of sentences on Σ, and as morphisms $\sigma: (\Sigma, \Gamma) \to (\Sigma', \Gamma')$ the signature morphisms $\sigma: \Sigma \to \Sigma'$ s.t. $Sen(\sigma)(\Gamma) \subseteq \Gamma'$. We use the functions $sign(\Sigma, \Gamma) = \Sigma$ and $ax(\Sigma, \Gamma) = \Gamma$ to select the signature and the axioms of a theory, respectively.

Since a set Γ of sentences on Σ determines the full subcategory of models that satisfy all the sentences Γ, it is easy to show that the functor Mod extends to a functor $Mod: \mathbf{Th}_0^{op} \to \mathbf{Cat}$. Similarly, by assigning to each theory $T = (\Sigma, \Gamma)$ the sentences on its signature we can extend the functor $Sen: \mathbf{Sign} \to \mathbf{Set}$ to a functor $Sen: \mathbf{Th}_0 \to \mathbf{Set}$.

Moreover, given an entailment system, and defining the set of theorems of a theory $T = (\Sigma, \Gamma)$ by $thm(T) = \{\phi \in Sen(\Sigma) \mid \Gamma \vdash_\Sigma \phi\}$, we then obtain a functor $thm: \mathbf{Th}_0 \to \mathbf{Set}$ that is a subfunctor of Sen. In the sequel we will find it convenient to extend the entailment relation from signatures to theories according to the following definition:

$$\Delta \vdash_{(\Sigma,\Gamma)} \phi \Leftrightarrow \Delta \cup \Gamma \vdash_\Sigma \phi.$$

[2] The definition of entailment system assumes that, as it is usually the case for most logics, the entailment relation is monotonic. To deal with entailment in nonmonotonic logics a weaker notion would be required.

Definition 3. A *logic* [24, Definition 6], is a 5-tuple $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$ such that:

(i) $(\mathbf{Sign}, Sen, \vdash)$ is an entailment system;

(ii) $(\mathbf{Sign}, Sen, Mod, \models)$ is an institution; and

(iii) the following *Soundness Condition* is satisfied: for each $\Sigma \in |\mathbf{Sign}|$, $\Gamma \subseteq Sen(\Sigma)$, and $\phi \in Sen(\Sigma)$, $\Gamma \vdash_\Sigma \phi \Rightarrow \Gamma \vdash_{\models_\Sigma} \phi$, where $\Gamma \vdash_{\models_\Sigma} \phi$ iff $[(M \models_\Sigma \gamma$ for all $\gamma \in \Gamma)$ implies $M \models_\Sigma \phi]$.

A logic is called *complete* if it satisfies the *Soundness and Completeness Condition*: for each $\Sigma \in |\mathbf{Sign}|$, $\Gamma \subseteq Sen(\Sigma)$, and $\phi \in Sen(\Sigma)$, $\Gamma \vdash_\Sigma \phi \Leftrightarrow \Gamma \vdash_{\models_\Sigma} \phi$.

A proof calculus associates to each theory T an "algebra of proofs" $P(T)$ in some adequate category of algebraic structures. From $P(T)$ the set *proofs*(T) of proofs derivable in the calculus is then obtained. The notion of logical system corresponds to the choice of a proof calculus for a logic.

Definition 4. A *proof calculus* [24, Definition 12], is a 6-tuple $\mathscr{P} = (\mathbf{Sign}, Sen, \vdash, P, Pr, \pi)$ such that:

(i) $(\mathbf{Sign}, Sen, \vdash)$ is an entailment system;

(ii) $P : \mathbf{Th}_0 \to \mathbf{Struct}_{\mathscr{P}}$ is a functor; for each theory T, the object $P(T) \in \mathbf{Struct}_{\mathscr{P}}$ is called its *proof-theoretic structure*;

(iii) $Pr : \mathbf{Struct}_{\mathscr{P}} \to \mathbf{Set}$ is a functor; for each theory T, the set $Pr(P(T))$ is called its *set of proofs*. Then *proofs* will denote the composite functor $Pr \cdot P : \mathbf{Th}_0 \to \mathbf{Set}$;

(iv) $\pi : proofs \Rightarrow Sen$ is a natural transformation, such that for each theory $T = (\Sigma, \Gamma)$, the image of $\pi_T : proofs(T) \to Sen(T)$ is the set $thm(T)$ of all sentences ϕ s.t. $\Gamma \vdash \phi$.

Definition 5. A *logical system* [24, Definition 12] $\mathscr{S} = (\mathbf{Sign}, Sen, Mod, \vdash, \models, P, Pr, \pi)$ is an 8-tuple such that:

(i) $(\mathbf{Sign}, Sen, Mod, \models, \vdash)$ is a logic;

(ii) $(\mathbf{Sign}, Sen, \vdash, P, Pr, \pi)$ is a proof calculus.

The crucial point of any categorical approach is the realization that the arrows between objects are more important than the objects themselves. In the context of the above definitions formalizing the different components of a logic, this means that we should look for adequate notions of mapping translating one logic into another; as we shall see, such mappings can be used to borrow components and tools from one logic to reuse them in another. In [24] maps of entailment systems, institutions, logics, proof calculi, and logical systems are defined and are illustrated with examples; here the definitions are presented in summarized form. Let us point out that maps of institutions differ from other notions of arrow between institutions, like the *institution morphisms* in [11] or the *simulations* in [1], mainly because each theory in the source institution is translated into a theory in the target institution

whose models represent (a subcategory of) the models of the starting theory. In Section 4 this mapping of theories from the source institution into the target institution will prove to be crucial in order to borrow a logic along a map of institutions.

Definition 6 ([24]). Given two entailment systems $\mathscr{E} = (\mathbf{Sign}, Sen, \vdash)$ and $\mathscr{E}' = (\mathbf{Sign}', Sen', \vdash')$, a *map of entailment systems* $(\Phi, \alpha) : \mathscr{E} \to \mathscr{E}'$ consists of a natural transformation $\alpha : Sen \Rightarrow Sen' \cdot \Phi$ and an α-sensible functor[3] $\Phi : \mathbf{Th}_0 \to \mathbf{Th}'_0$ satisfying the following property:

$$\Gamma \vdash_\Sigma \phi \;\Rightarrow\; \Gamma' \cup \alpha_\Sigma(\Gamma) \vdash'_{\Sigma'} \alpha_\Sigma(\phi),$$

where, by convention, $(\Sigma', \Gamma') = \Phi(\Sigma, \emptyset)$. Using the property of α-sensible functors mentioned in the footnote, we can rephrase this requirement in a simpler and more compact way as follows:

$$\Gamma \vdash_\Sigma \phi \;\Rightarrow\; \alpha_\Sigma(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_\Sigma(\phi).$$

We call the map (Φ, α) *conservative* if in addition we have

$$\Gamma \vdash_\Sigma \phi \;\Leftrightarrow\; \alpha_\Sigma(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_\Sigma(\phi).$$

Given institutions $\mathscr{I} = (\mathbf{Sign}, Sen, Mod, \models)$ and $\mathscr{I}' = (\mathbf{Sign}', Sen', Mod', \models')$, a *map of institutions* $(\Phi, \alpha, \beta) : \mathscr{I} \to \mathscr{I}'$ consists of a natural transformation $\alpha : Sen \Rightarrow Sen' \cdot \Phi$, an α-sensible[4] functor $\varphi : \mathbf{Th}_0 \to \mathbf{Th}'_0$, and a natural transformation $\beta : Mod' \cdot \Phi^{op} \Rightarrow Mod$ such that for each $\Sigma \in |\mathbf{Sign}|$, each $\phi \in Sen(\Sigma)$, and each $M' \in |Mod'(\phi(\Sigma, \emptyset))|$ the following property is satisfied:

$$M' \models'_{\Sigma'} \alpha_\Sigma(\phi) \;\Leftrightarrow\; \beta_{(\Sigma,\emptyset)}(M') \models_\Sigma \phi,$$

where Σ' is the signature of the theory $\Phi(\Sigma, \emptyset)$.

Given logics $\mathscr{L}, \mathscr{L}'$, a *map of logics* $(\Phi, \alpha, \beta) : \mathscr{L} \to \mathscr{L}'$ is a map $(\Phi, \alpha, \beta) : inst(\mathscr{L}) \to inst(\mathscr{L}')$ of the underlying institutions such that, in addition, $(\Phi, \alpha) : ent(\mathscr{L}) \to ent(\mathscr{L}')$ is a map of the underlying entailment systems.

Given proof calculi $\mathscr{P}$ and $\mathscr{P}'$, a map of proof calculi $(\Phi, \alpha, \gamma) : \mathscr{P} \to \mathscr{P}'$ consists of a map $(\Phi, \alpha) : ent(\mathscr{P}) \to ent(\mathscr{P}')$ of the underlying entailment systems together with a natural transformation $\gamma : proofs \Rightarrow proofs' \cdot \Phi$ such that $\pi'_\Phi \circ \gamma = \alpha \circ \pi$.

---

[3] We refer to [24] for the detailed definition of α-sensible functors. Basically, what is required is that the provable consequences of the theory $\Phi(\Sigma, \Gamma)$ are entirely determined by $\Phi(\Sigma, \emptyset)$ and by α; specifically, we must have the condition

$$thm'(\Phi(\Sigma, \Gamma)) = thm'(\Sigma', \Gamma' \cup \alpha(\Gamma)),$$

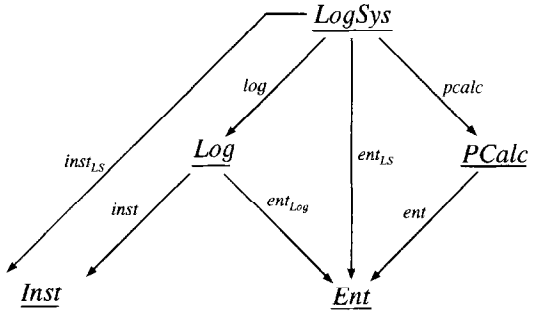
where, by convention, $(\Sigma', \Gamma') = \Phi(\Sigma, \emptyset)$. By allowing functors Φ sending theories to theories, instead of more restrictive functors sending signatures to signatures, we can define much more flexible mappings between logics.

[4] Φ is α-sensible relative to the "logical consequence" entailment relations $\vdash_{\models}$ and $\vdash_{\models'}$ associated to $\mathscr{I}$ and $\mathscr{I}'$, where, by definition, $\Gamma \vdash_{\models_\Sigma} \phi$ iff $(M \models_\Sigma \gamma$ for all $\gamma \in \Gamma)$ implies $M \models_\Sigma \phi$.

Given logical systems $\mathscr{S}$ and $\mathscr{S}'$, a *map of logical systems* $(\Phi, \alpha, \beta, \gamma) : \mathscr{S} \to \mathscr{S}'$ consists of a map of the underlying logics $(\Phi, \alpha, \beta) : log(\mathscr{S}) \to log(\mathscr{S}')$ and a map of the underlying proof calculi $(\Phi, \alpha, \gamma) : pcalc(\mathscr{S}) \to pcalc(\mathscr{S}')$.

We denote by *Ent* the category[5] of entailment systems with maps of entailment systems as arrows, by *Inst* the category of institutions with maps of institutions as arrows, by *Log* the category of logics with maps of logics as arrows, by *PCalc* the category of proof calculi with maps of proof calculi as arrows, and by *LogSys* the category of logical systems with maps of logical systems as arrows.

The relationships among the categories of entailment systems, institutions, logics, proof calculi, and logical systems can be illustrated by the following diagram, where all the arrows depicted are forgetful functors that "throw away" appropriate components of a logic. For example, the functor *inst* maps a logic $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$ to the institution $(\mathbf{Sign}, Sen, Mod, \models)$. As further explained below, both *log* and *ent* have right adjoints, corresponding to regarding theorems as proofs of themselves, ent_{Log} and *pcalc* have right adjoints, obtained by adding the empty set of models for each signature, and *inst* has both a left and right adjoint, obtained by adding, respectively, the set membership and the validity relations as entailment. Therefore, all functors in the diagram have right adjoints.



Proposition 7 ([24, Proposition 31]). *The functor* $(\_)^+ : \underline{Inst} \to \underline{Log}$, *defined by*

- $(\mathscr{I})^+ = (\mathbf{Sign}, Sen, Mod, \models, \vdash_{\models})$, where $\Gamma \vdash_{\models} \phi$ iff $(M \models_\Sigma \gamma$ for all $\gamma \in \Gamma)$ *implies* $M \models_\Sigma \phi$, *for any institution* $\mathscr{I} = (\mathbf{Sign}, Sen, Mod, \models)$, *and*
- $\rho^+ = \rho$ *for any map* ρ,

[5] Since in *Ent* (and similarly in the other categories discussed below) the objects of the category of signatures may form a class, and not just a set, the collection of maps between two entailment systems may likewise form a class; however, if signatures, sentences, models, and so on are required to belong to a suitable universe, that we never mention, as usual, the well-known foundational problems arising whenever one speaks of the *category of all categories* are avoided. For similar remarks about foundations see also [13, 24, 31].

is right adjoint to the forgetful functor inst : Log → Inst. Moreover, the unit of the adjunction for a logic $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$ *is the map* $(Id_{\mathbf{Sign}}, Id_{Sen}, Id_{Mod})$ *and the counit is the identity natural transformation.*

Proposition 8 ([24, Proposition 31]). *The functor* $(\_)^- : \underline{Inst} \to \underline{Log}$, *defined by*
- $(\mathscr{I})^- = (\mathbf{Sign}, Sen, Mod, \models, \vdash^-)$, *where* $\Gamma \vdash^- \phi$ *iff* $\phi \in \Gamma$, *for any institution* $\mathscr{I} = (\mathbf{Sign}, Sen, Mod, \models)$, *and*
- $\rho^- = \rho$ *for any map* ρ,

is left adjoint to the forgetful functor inst. Moreover, the unit of the adjunction for an institution $\mathscr{I}$ *is the identity of* $\mathscr{I}$.

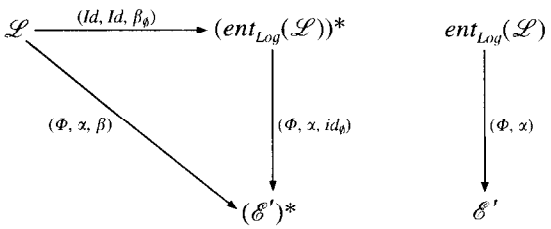
Adding to any entailment system the empty model component and the empty validity relation we define a functor, that is the right adjoint to the forgetful functor from logics to entailment systems.

Proposition 9. *The functor* $(\_)^* : \underline{Ent} \to \underline{Log}$, *defined by*
- *for each entailment system* $\mathscr{E} = (\mathbf{Sign}, Sen, \vdash)$ *let* $(\mathscr{E})^*$ *be the logic* $(\mathbf{Sign}, Sen, Mod_\emptyset, \models_\emptyset, \vdash)$, *where* $Mod_\emptyset$ *is the functor mapping each signature to the empty category of models and each map of signatures to the empty identity map, and where* $\models_\emptyset$ *assigns to each signature the empty satisfaction relation*;
- *for each map of entailment systems* (Φ, α) *let* $(\Phi, \alpha)^*$ *be* $(\Phi, \alpha, id_\emptyset)$, *where* $id_\emptyset$ *is the natural transformation with all components of the form* $Id_\emptyset : \emptyset \to \emptyset$;

is right adjoint to the functor $ent_{Log} : \underline{Log} \to \underline{Ent}$. *Moreover, the unit of the adjunction for a logic* $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$ *is the map* $(Id_{\mathbf{Sign}}, Id_{Sen}, \beta_\emptyset)$, *where for each theory* (Σ, Γ) $\beta_{\emptyset(\Sigma,\Gamma)}$ *is the empty function* $\emptyset \to Mod(\Sigma, \Gamma)$, *and the counit is the identity natural transformation.*

Proof. Let $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$ be a logic, let $\mathscr{E}' = (\mathbf{Sign}', Sen', \vdash')$ be an entailment system, and let $(\Phi, \alpha, \beta) : \mathscr{L} \to (\mathscr{E}')^*$ be a map of logics.

Then, since for each theory (Σ, Γ) we have $\beta_{(\Sigma,\Gamma)} : Mod'_\emptyset(\Phi(\Sigma, \Gamma)) \to Mod(\Sigma, \Gamma)$, and $Mod'_\emptyset(\Phi(\Sigma, \Gamma)) = \emptyset$, we have $Mod'_\emptyset \cdot \Phi^{op} = Mod_\emptyset$, and therefore $\beta = \beta_\emptyset$. As a consequence, $(\Phi, \alpha) : ent_{Log}(\mathscr{L}) \to \mathscr{E}'$ is the unique map of entailment systems such that $(\Phi, \alpha)^*$ makes the following diagram commute.



By definition of $(\_)^*$, the composition $ent_{Log} \cdot (\_)^*$ is the identity $Id_{\underline{Ent}}$; and it is immediate, by checking the corresponding diagram with the counit, that the counit is the identity. □

Analogously, adding the empty model component to proof calculi, we get a right adjoint to the forgetful functor from logical systems to proof calculi.

Proposition 10. *The functor* $(\_)^{\triangle} : \underline{PCalc} \to \underline{LogSys}$, *defined by*
- *for each proof calculus* $\mathscr{P} = (\mathbf{Sign}, Sen, \vdash, P, Pr, \pi)$ *let* $(\mathscr{P})^{\triangle}$ *be the logical system* $(\mathbf{Sign}, Sen, Mod_{\emptyset}, \vdash, \models_{\emptyset}, P, Pr, \pi)$, *where* $Mod_{\emptyset}$ *is the functor mapping each signature to the empty category of models*;
- *for each map of proof calculi* (Φ, α, γ) *let* $(\Phi, \alpha, \gamma)^{\triangle}$ *be* $(\Phi, \alpha, id_{\emptyset}, \gamma)$, *where* $id_{\emptyset}$ *is the natural transformation with all components the empty identity map*;

is right adjoint to the functor pcalc : $\underline{LogSys} \to \underline{PCalc}$. *Moreover, the unit of the adjunction for a logical system* $\mathscr{S} = (\mathbf{Sign}, Sen, Mod, \vdash, \models, P, Pr, \pi)$ *is the map* $(Id_{\mathbf{Sign}}, Id_{Sen}, \beta_{\emptyset}, Id_{proofs})$ *and the counit is the identity natural transformation.*

Proof. Analogous to the proof of Proposition 9. □

Proposition 11 ([24, Proposition 34]). *The functor* $(\_)^{\sharp} : \underline{Ent} \to \underline{PCalc}$, *defined by*
- $(\mathbf{Sign}, Sen, \vdash)^{\sharp} = (\mathbf{Sign}, Sen, \vdash, thm, Id_{\mathbf{Set}}, j)$, *where thm is the functor sending each theory to the set of its theorems, described in Section* 2.2 *of* [24], *and j is the natural subfunctor inclusion* $thm \subseteq Sen$, *and*
- $(\Phi, \alpha)^{\sharp} = (\Phi, \alpha, \alpha_{|thm})$, *where* $\alpha_{|thm}$ *is the restriction of* α *to the theorems of the domain, for any map* (Φ, α),

is right adjoint to the forgetful functor ent : $\underline{PCalc} \to \underline{Ent}$. *Moreover, the unit of the adjunction for a proof calculus* $\mathscr{P} = (\mathbf{Sign}, Sen, \vdash, P, Pr, \pi)$ *is the map* $(Id_{\mathbf{Sign}}, Id_{Sen}, \pi)$, *where* π *is now viewed as a natural transformation* $\pi : proofs \Rightarrow thm$, *and the counit is the identity natural transformation.*

Proposition 12 ([24, Proposition 37]). *The functor* $(\_)^{\diamond} : \underline{Log} \to \underline{LogSys}$, *defined by*
- $(\mathscr{L})^{\diamond} = (\mathbf{Sign}, Sen, Mod, \vdash, \models, thm, Id_{\mathbf{Set}}, j)$, *for any logic* $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$, *and*
- $(\Phi, \alpha, \beta)^{\diamond} = (\Phi, \alpha, \beta, \alpha_{|thm})$, *where* $\alpha_{|thm}$ *is the restriction of* α *to the theorems of the domain, for any map* (Φ, α, β),

is right adjoint to the forgetful functor log : $\underline{LogSys} \to \underline{Log}$. *Moreover, the unit of the adjunction for a logical system* $\mathscr{S} = (\mathbf{Sign}, Sen, Mod, \vdash, \models, P, Pr, \pi)$ *is the map* $(Id_{\mathbf{Sign}}, id_{Sen}, Id_{Mod}, \pi)$, *where* π *is now viewed as a natural transformation* π : *proofs* $\Rightarrow$ *thm, and the counit is the identity natural transformation.*

Proposition 13 ([24, Propositions 31 and 37]). *The functor* $(\_)^{\flat} : \underline{Inst} \to \underline{LogSys}$, *defined by*

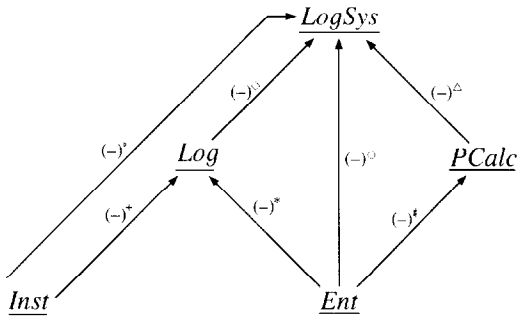
- $(\mathscr{I})^{\flat} = (\mathbf{Sign}, Sen, Mod, \vdash_{\models}, \models, thm, Id_{\mathbf{Set}}, j)$, *for any institution* $\mathscr{I} = (\mathbf{Sign}, Sen, Mod, \models)$, *and*
- $(\Phi, \alpha, \beta)^{\flat} = (\Phi, \alpha, \beta, \alpha_{|thm})$, *for any map* (Φ, α, β),

is right adjoint to the forgetful functor $inst_{LS} : \underline{LogSys} \to \underline{Inst}$. *Moreover, the unit of the adjunction for a logical system* $\mathscr{S} = (\mathbf{Sign}, Sen, Mod, \vdash, \models, P, Pr, \pi)$ *is the map* $(Id_{\mathbf{Sign}}, Id_{Sen}, Id_{Mod}, \pi)$ *and the counit is the identity natural transformation.*

Proposition 14 ([24, Propositions 31 and 37]). *The functor* $(\_)^{\circ} : \underline{Ent} \to \underline{LogSys}$, *defined by* $(\_)^{\circ} = (\_)^{\triangle} \cdot (\_)^{\sharp} = (\_)^{\diamond} \cdot (\_)^{*}$ *is the right adjoint to the forgetful functor* $ent_{LS} : \underline{LogSys} \to \underline{Ent}$. *Moreover, the unit of the adjunction for a logical system* $\mathscr{S} = (\mathbf{Sign}, Sen, Mod, \vdash, \models, P, Pr, \pi)$ *is the map* $(Id_{\mathbf{Sign}}, Id_{Sen}, \beta_{\emptyset}, \pi)$ *and the counit is the identity natural transformation.*

Proof. By definition $(\_)^{\triangle} \cdot (\_)^{\sharp} = (\_)^{\diamond} \cdot (\_)^{*}$; moreover, since the composition of adjoint situations yields an adjoint situation, too, by Propositions 10 and 11 (equivalently by Propositions 12 and 9) we have the thesis. □

Since the counits of the right adjoints to the above forgetful functors are all identities, the right adjoints are all full and faithful (see e.g. [20, IV.3]) and injective on the objects. Therefore they are, up to isomorphism, reflective subcategory inclusions. That means that all the arrows in the following diagram can be seen as ways of wrapping poorer logical structures in order to embed them as full subcategories of categories of richer logical structures.



3. Extension and fibration

The transportation of structure that we study will involve two categories of logical structures, **C** and **D**, and a couple of functors $U : \mathbf{D} \to \mathbf{C}$ and $R : \mathbf{C} \to \mathbf{D}$ with R

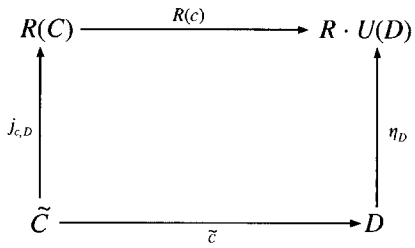


Fig. 1. Pullback diagram.

right adjoint to U. In the applications that we will consider, U, in spite of being a left adjoint, will have the flavor of a forgetful functor.[6] The basic construction applies to arbitrary categories $\mathbf{C}$ and $\mathbf{D}$ with functors U and R as above, and consists in the process of enriching an object $C \in |\mathbf{C}|$ with the structure of an object $D \in |\mathbf{D}|$ via a map $c : C \to U(D)$. The transported structure is enjoyed by an object $\tilde{C} \in |\mathbf{D}|$ which, roughly speaking, enriches C with the features of D translated by c. Formally, it is the pullback diagram shown in Fig. 1, where η_D is the unit of the adjunction for the object D; therefore, the construction is strongly dependent on c and D. Moreover, the construction is functorial, in the sense that it preserves the c-consistent translations of C and D. Indeed it is a functor between two comma categories, associating arrows in $\mathbf{C}$ of the form $c : C \to U(D)$, i.e. objects in $\mathbf{C} \downarrow U$, with arrows in $\mathbf{D}$, i.e. objects in $\mathbf{D}^{\to} = \mathbf{D} \downarrow \mathbf{D}$. It is also worth noting that this association is optimal in the sense that it is the right adjoint of the obvious lifting of U to $\mathbf{C} \downarrow U$. The details of the construction as an adjointness are given in Theorem 16 below. Later in this section we give general conditions under which this construction makes the functor $U : \mathbf{D} \to \mathbf{C}$ a *fibration*, so that $\tilde{c}$ is a cartesian lifting of the map c.

Definition 15. Let $\mathbf{C}$ and $\mathbf{D}$ be categories and let $U : \mathbf{D} \to \mathbf{C}$ be a functor. Let $\mathbf{D}^{\to}$ denote the comma category $\mathbf{D} \downarrow \mathbf{D}$ (see e.g. [20]) and let $\mathbf{C}^{\downarrow U}$ denote the comma category $\mathbf{C} \downarrow U$. Then U induces a functor $\vec{U} : \mathbf{D}^{\to} \to \mathbf{C}^{\downarrow U}$, defined as follows:

- $\vec{U}(d : D \to D') = (U(d) : U(D) \to U(D'), D')$ for each object $(d : D \to D') \in |\mathbf{D}^{\to}|$;
- $\vec{U}(f, f') = (U(f), f')$ for each arrow (f, f') in $\mathbf{D}^{\to}$ from $(d_1 : D_1 \to D_1')$ into $(d_2 : D_2 \to D_2')$.

Let us now show that if a functor $U : \mathbf{D} \to \mathbf{C}$ has a right adjoint $R : \mathbf{C} \to \mathbf{D}$, then R can be lifted up to a functor $\vec{R} : \mathbf{C}^{\downarrow U} \to \mathbf{D}^{\to}$, right adjoint to $\vec{U}$, if the appropriate pullbacks in $\mathbf{D}$ exist.

[6] This is of course somewhat counterintuitive, but it holds for all the forgetful functors that throw away logical structure discussed in Section 2; however, U may in some cases – as for example for $U = inst$ in the first diagram in Section 2 – have also a left adjoint.

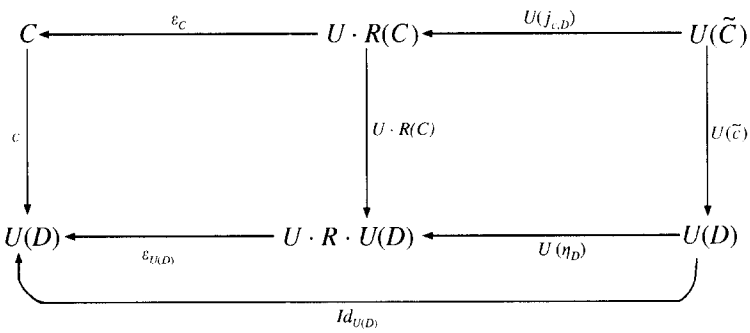
Theorem 16. *Let* $\mathbf{C}$ *and* $\mathbf{D}$ *be categories, let* $R : \mathbf{C} \to \mathbf{D}$ *be right adjoint to* $U : \mathbf{D} \to \mathbf{C}$, *with* $\eta : 1_{\mathbf{D}} \Rightarrow R \cdot U$ *the unit and* $\varepsilon : U \cdot R \Rightarrow 1_{\mathbf{C}}$ *the counit of the adjunction. If for each object* $C \in |\mathbf{C}|$, *each object* $D \in |\mathbf{D}|$, *and each arrow* $c \in \mathbf{C}(C, U(D))$ *the pullback of* $R(c)$ *and* η_D *in Fig.* 1 *exists, then the functor* $\vec{U}$ *has a right adjoint* $\vec{\vec{R}} : \mathbf{C}^{\downarrow U} \to \mathbf{D}^{\to}$, *and the counit of this adjunction is the family* $\{(\varepsilon_C \cdot U(j_{c,D}), Id_D) \mid (c : C \to U(D), D) \in |\mathbf{C}^{\downarrow U}|\}$.

We then say that $\mathbf{C}$ admits extension under R and U, *and for each arrow* $(c : C \to U(D), D)$ *in* $\mathbf{C}$ *the arrow* $\tilde{c} : \tilde{C} \to D$ *in* $\mathbf{D}$ *is called* the extension of c *by* R and U.

Proof. For a right adjoint $\vec{\vec{R}}$ of $\vec{U}$ to exist, it is sufficient to show that for each object $(c : C \to U(D), D)$ in $\mathbf{C}^{\downarrow U}$ there exist an object $\vec{\vec{R}}(c, D)$ in $\mathbf{D}^{\to}$ and a co-universal arrow $\alpha_{c,D} : \vec{U}(\vec{\vec{R}}(c, D)) \to (c, D)$ from $\vec{U}(\vec{\vec{R}}(c, D))$ to (c, D).

Let us define $\vec{\vec{R}}(c, D) = \tilde{c}$, where $\tilde{c}$ is the pullback of c along η_D, given in Fig. 1.

It is easy to check that $\alpha_{c,D} = (\varepsilon_C \cdot U(j_{c,D}), Id_D)$ is an arrow from $\vec{U} \cdot \vec{\vec{R}}(c, D)$ to (c, D). Indeed, let us consider the following diagram, that is commutative, since it is the pasting of the translation along U of the pullback defining $\vec{\vec{R}}(c, D)$ with the diagram expressing the naturality of ε.



To show that $\alpha_{c,D}$ is co-universal, let us fix an object $d : D_0 \to D_1$ in $\mathbf{D}^{\to}$ and an arrow $(f, g) : \vec{U}(d) \to (c, D)$ in $\mathbf{C}^{\downarrow U}$, and let us show that a unique arrow $\phi : d \to \vec{\vec{R}}(c, D)$ exists s.t. $\alpha_{c,D} \cdot \vec{U}(\phi) = (f, g)$.

Since (f, g) is an arrow from $\vec{U}(d) = (U(d) : U(D_0) \to U(D_1), D_1)$ to $c : C \to U(D), c \cdot f = U(g) \cdot U(d)$, and hence $R(c \cdot f) \cdot \eta_{D_0} = R(U(g) \cdot U(d)) \cdot \eta_{D_0}$. So, by the naturality of $\eta, R(c \cdot f) \cdot \eta_{D_0} = \eta_D \cdot g \cdot d$; and hence, by the pullback definition of $\tilde{C}$, there exists a unique $\xi : D_0 \to \tilde{C}$ s.t.

(i) $\tilde{c} \cdot \xi = g \cdot d$;

(ii) $j_{c,D} \cdot \xi = R(f) \cdot \eta_{D_0}$.

The first condition is equivalent to $\phi = (\xi, g)$ being an arrow from d to $\vec{\vec{R}}(c, D)$ in $\mathbf{D}^{\to}$. From the second one and the naturality of ε, we have $\varepsilon_C \cdot U(j_{c,D}) \cdot U(\xi) =$

$\varepsilon_C \cdot U(R(f)) \cdot U(\eta_{D_0}) = f \cdot \varepsilon_{U(D_0)} \cdot U(\eta_{D_0})$; but $\varepsilon_{U(D_0)} \cdot U(\eta_{D_0})$ is the identity, and hence $\varepsilon_C \cdot U(j_{c,D}) \cdot U(\xi) = f$.

Therefore $\alpha_{c,D} \cdot \vec{U}(\xi, g) = (\varepsilon_C \cdot U(j_{c,D}) \cdot U(\xi), Id_D \cdot g) = (f, g)$. Let us show that $\phi = (\xi, g)$ is unique; let us assume that $\alpha_{c,D} \cdot \vec{U}(\phi') = (f, g)$ for some $\phi' = (\xi' g')$ and let us then show that $\phi' = \phi$. Since the second component of $\alpha_{c,D}$ is the identity, it follows immediately that $g' = g$; in order to show that $\xi' = \xi$ we prove that ξ' is a factorization for the maps $g \cdot d$, and $R(f) \cdot \eta_{D_0}$ through the pullback $\tilde{C}$ as ξ is, and hence, by the uniqueness of the factorization through the pullback, we have $\xi' = \xi$. As (ξ', g') is an arrow in $\mathbf{D}^{\rightarrow}$ from d into $\vec{R}(c, D)$, we have $\tilde{c} \cdot \xi' = g \cdot d$. Since $\alpha_{c,D} \cdot \vec{U}(\xi', g') = (f, g), \varepsilon_C \cdot U(j_{c,D}) \cdot U(\xi') = f$ hence $R(f) \cdot \eta_{D_0} = R(\varepsilon_C) \cdot (R \cdot U(j_{c,D} \cdot \xi')) \cdot \eta_{D_0}$. Thus, by naturality of η, we get $R(f) \cdot \eta_{D_0} = R(\varepsilon_C) \cdot \eta_{R(C)} \cdot j_{c,D} \cdot \xi' = j_{c,D} \cdot \xi'$, where the last identity holds because $R(\varepsilon) \circ \eta_R$ is the identity. □

We next show that, just as the adjoint functors compose to yield another adjoint functor, two extension processes of the kind described also yield an extension process by composition.

Proposition 17. *Let* $R_1 : \mathbf{C} \to \mathbf{D}$ *be the right adjoint of* $U_1 : \mathbf{D} \to \mathbf{C}$, *and let* $R_2 : \mathbf{D} \to \mathbf{E}$ *be the right adjoint of* $U_2 : \mathbf{E} \to \mathbf{D}$.

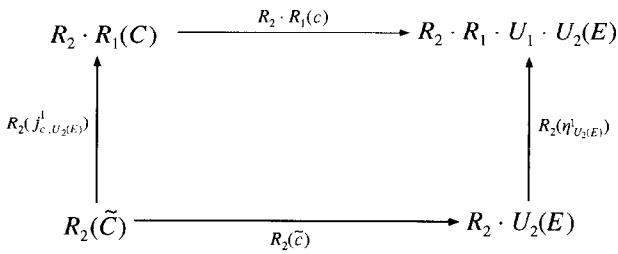
If $\mathbf{C}$ *admits extension under* R_1 *and* U_1, *and* $\mathbf{D}$ *admits extension under* R_2 *and* U_2, *then* $\mathbf{C}$ *admits extension under* $R_2 \cdot R_1$ *and* $U_1 \cdot U_2$.

Moreover, the extension of any $c : C \to U_1 \cdot U_2(E)$ *by* $R_2 \cdot R_1$ *and* $U_1 \cdot U_2$ *is* $\hat{\tilde{c}} : \hat{\tilde{C}} \to E$, *where* $\tilde{c} : \tilde{C} \to D$ *denotes the extension of any* $c : C \to U_1(D)$ *by* U_1 *and* R_1, *and* $\hat{d} : \hat{D} \to E$ *denotes the extension of any* $d : D \to U_2(E)$ *by* U_2 *and* R_2.

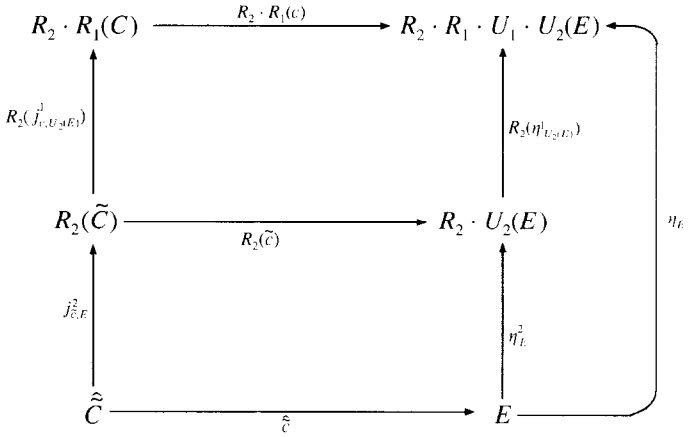
Proof. Let us denote by η^1 the unit of the adjunction between R_1 and U_1 and by η^2 the unit of the adjunction between R_2 and U_2; then $R_2 \cdot R_1$ is the right adjoint of $U_1 \cdot U_2$, with unit $\eta = R_2(\eta^1_{U_2}) \cdot \eta^2$ (see e.g. [18, Proposition 27.8]).

Let us consider $c : C \to U_1 \cdot U_2(E)$ and let us show that there exists the pullback of $R_2 \cdot R_1(c)$ along η_E.

Since $\mathbf{C}$ admits extension under R_1 and U_1, the pullback of $R_1(c)$ along $\eta^1_{U_2(E)}$ exists and hence, as right adjoints preserve pullbacks, the following diagram is a pullback:



Since **D** admits extension under R_2 and U_2, the pullback of $R_2(\tilde{c})$ along η^2_E exists and hence, as pasting pullback squares together gives a pullback too, the following diagram is a pullback:



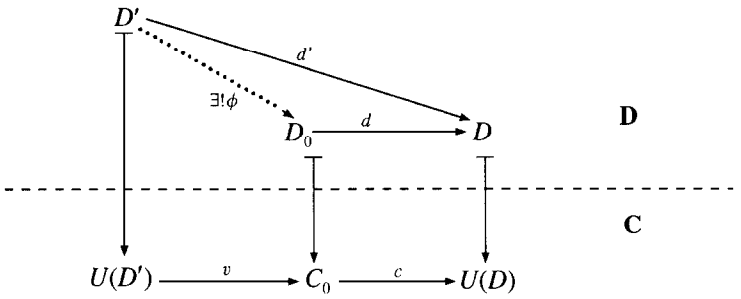
Therefore the hypothesis of Theorem 16 is satisfied, and hence **C** admits extension under $R_2 \cdot R_1$ and $U_1 \cdot U_2$. □

In general, the extension $\tilde{c} : \tilde{C} \to D$ built by the adjoint construction in Theorem 16 for a map $c : C \to U(D)$ does not lie *above* c, in the sense that $U(\tilde{c})$ is in general different from c; it could be not even isomorphic. However, if $\tilde{c}$ is above c for each possible c in a natural way, then our construction corresponds to a cartesian lifting of c, and U is a fibration. In fact, using a characterization of fibrations in terms of comma categories due to Chevalley, the very close relationship between fibrations and the above extension construction becomes particularly clear. In addition, we give in Corollary 22 sufficient conditions – satisfied in all the applications to transportation of logical structure discussed in Section 4 – under which an extension process is actually a fibration.

Definition 18. Let $U : \mathbf{D} \to \mathbf{C}$ be a functor; a morphism $d : D_0 \to D$ in **D** is called *cartesian* over a morphism $c : C_0 \to C$ in **C** if $U(d) = c$ and for each $d' : D' \to D$ in **D** with $U(d') = c \cdot v$ there exists a unique $\phi : D' \to D_0$ in **D** s.t. $U(\phi) = v$ and $d \cdot \phi = d'$.

The functor U is called a *fibration* if for each $D \in \mathbf{D}$ and each morphism $c : C_0 \to U(D)$ in **C** there exists a cartesian morphism over c with codomain D.

The cartesian lifting described in the last definition can be graphically represented by the following diagram.



The "Chevalley criterion" states that given a functor $U : \mathbf{D} \to \mathbf{C}$ the existence of a right adjoint $\overrightarrow{R} : \mathbf{C}^{\downarrow U} \to \mathbf{D}^{\to}$ to $\overrightarrow{U} : \mathbf{D}^{\to} \to \mathbf{C}^{\downarrow U}$ s.t. the counit $\alpha : \overrightarrow{U} \cdot \overrightarrow{R} \Rightarrow 1_{\mathbf{C}^{\downarrow U}}$ is the identity is equivalent to U being a fibration.

Theorem 19 (Chevalley criterion [3]). *Let $U : \mathbf{D} \to \mathbf{C}$ be a functor. Then the following conditions are equivalent:*

- *$\overrightarrow{U}$ has a right adjoint $\overrightarrow{R} : \mathbf{C}^{\downarrow U} \to \mathbf{D}^{\to}$ s.t. $\overrightarrow{U} \cdot \overrightarrow{R}$ is the identity and the counit of the adjunction is the identity natural transformation;*
- *U is a fibration.*

Proof. See Proposition 3.11 in [16]. □

The above characterization of fibrations in terms of comma categories yields as an easy corollary a condition under which the extension construction yields a fibration.

Corollary 20. *Using the notation of Theorem* 16, *let* **C** *admit extension under R and U; if $\overrightarrow{U} \cdot \overrightarrow{R}$ is the identity and the counit of the adjunction between $\overrightarrow{U}$ and $\overrightarrow{R}$ is the identity natural transformation, then U is a fibration.*

Proof. By Theorem 19, that applies because of Theorem 16. □

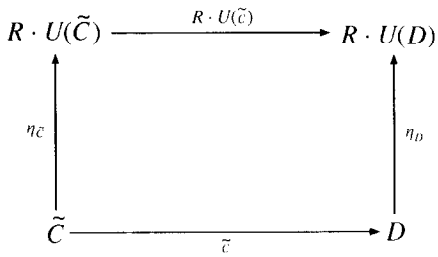
Note that, since in the adjointness formulation of fibrations given by the Chevalley criterion the right adjoint $\overrightarrow{R}$ sends each map in $\mathbf{C}^{\downarrow U}$ to a cartesian lifting, it follows immediately under the assumptions of Corollary 20 that then the extension map $\tilde{c} : \tilde{C} \to D$ is a cartesian lifting of the map $(c : C \to U(D), D)$.

All the borrowing constructions that we shall discuss in Section 4 correspond to fibrations that are obtained by extension constructions satisfying the conditions in Corollary 20. In addition, the extension constructions that we shall encounter all have a particularly nice and simple description as *cartesian reflections* in the sense of the following

Definition 21. A pair of functors $U : \mathbf{D} \to \mathbf{C}$ and $R : \mathbf{C} \to \mathbf{D}$, with R right adjoint to U, is called a *cartesian reflection* iff:

(i) $U \cdot R$ is the identity functor on $\mathbf{C}$, and the counit ε is the identity natural transformation; and

(ii) for each map $c : C \to U(D)$ in $\mathbf{C}$ there is a map $\tilde{c} : \tilde{C} \to D$ in $\mathbf{D}$ such that $U(\tilde{c}) = c$ and the commutative square



is a pullback.

Note that, since $R \cdot U(\tilde{c}) = R(c)$ and $R \cdot U(\tilde{C}) = R(C)$, if U, R form a cartesian reflection, then $\mathbf{C}$ admits extension under R and U, since the above pullback provides the extension construction with $j_{c,D} = \eta_{\tilde{C}}$. Moreover,

Corollary 22. *If the functors $U : \mathbf{D} \to \mathbf{C}$ and $R : \mathbf{C} \to \mathbf{D}$ form a cartesian reflection, then U is a fibration.*

Proof. It is sufficient to show that the conditions in Corollary 20 are satisfied. By Theorem 16 specialized to this case, in which $j_{c,D} = \eta_{\tilde{C}}$, the counit of the adjunction between $\overrightarrow{U}$ and $\overrightarrow{R}$ for each object $(c : C \to U(D), D)$ is $\alpha_{c,D} = (\varepsilon_C \cdot U(\eta_{\tilde{C}}), Id_D)$.

Since by hypothesis ε is the identity, and by the adjunction equations, $\varepsilon_U \circ U(\eta)$ is the identity, $U(\eta)$ is also the identity. Therefore, $\alpha_{c,D}$ is the identity natural transformation, as desired. □

Being a cartesian reflection is a particularly good property, since checking that a map is cartesian lifting is reduced to checking that the naturality diagram for η is a pullback. Moreover, cartesian reflections have additional properties not enjoyed by arbitrary fibrations.[7]

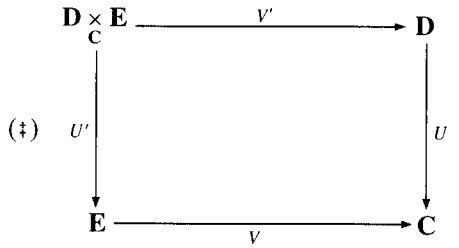
It is well known that fibrations are closed under composition, and that the pullback of a fibration is also a fibration [16, 19]. Cartesian reflections also enjoy these properties.

[7] For example, the functor R ensures the existence of fibered terminal objects in the fibration (therefore each $R(C)$ is a terminal object in the fiber category $\mathbf{D}_C$ of objects $D \in |\mathbf{D}|$ with $U(D) = C$).

Proposition 23. *Let* $U_1 : \mathbf{D} \to \mathbf{C}$, $R_1 : \mathbf{C} \to \mathbf{D}$, *and* $U_2 : \mathbf{E} \to \mathbf{D}$, $R_2 : \mathbf{D} \to \mathbf{E}$ *be cartesian reflections. Then* $U_1 \cdot U_2$, $R_2 \cdot R_1$ *is a cartesian reflection.*

Proof. The proof follows easily from the proof of Proposition 17 using the adjunction equations and the additional fact that the counits are identities. □

Proposition 24. *Let* $U : \mathbf{D} \to \mathbf{C}$, $R : \mathbf{C} \to \mathbf{D}$ *be a cartesian reflection, and let* $V : \mathbf{E} \to \mathbf{C}$ *be any functor. Then, in the pullback square*

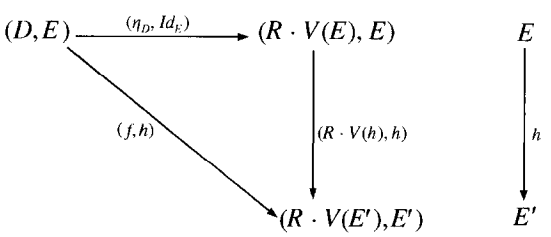


The functor U' *has a right adjoint* R' *such that* U', R' *is a cartesian reflection.*

Proof. Recalling that the pullback $\mathbf{D} \underset{\mathbf{C}}{\times} \mathbf{E}$ is the subcategory of $\mathbf{D} \times \mathbf{E}$ with objects the pairs (D,E) s.t. $U(D) = V(E)$, with morphisms those (f,g) s.t. $U(f) = V(g)$, and with $V'(U')$ the first (second) projections, define R' by:
- $R'(E) = (R \cdot V(E), E)$ for all $E \in |\mathbf{E}|$;
- $R'(h) = (R \cdot V(h), h)$ for all $h : E \to E'$ in $\mathbf{E}$.

Then, choose as unit maps the $\eta'_{(D,E)} = (\eta_D, Id_E) : (D,E) \to (R \cdot V(E), E)$, which is a well-defined arrow in $\mathbf{D} \underset{\mathbf{C}}{\times} \mathbf{E}$, since by the adjunction equations and U, R a cartesian reflection we have $U(\eta_D) = Id_{U(D)}$, which is equal to $V(Id_E) = Id_{V(E)} = Id_{U(D)}$, given that $(D,E) \in |\mathbf{D} \underset{\mathbf{C}}{\times} \mathbf{E}|$ implies $U(D) = V(E)$.

Consider now the diagram



Then there is at most one map $g : E \to E'$ such that $R'(g) \cdot \eta'_{(D,E)} = (f,h)$, namely $g = h$, since we have

$$R'(g) \cdot \eta'_{(D,E)} = (R \cdot V(g), g) \cdot (\eta_D, Id_E) = (R \cdot V(g) \cdot \eta_D, g) = (f,h)$$

and the last equality forces $g = h$. Therefore, to prove the universal property we only have to check the equation $R \cdot V(h) \cdot \eta_D = f$, which by $V(h) = U(f)$, is equivalent to the equation $R \cdot U(f) \cdot \eta_D = f$, that by the naturality of η reduces to the equation $\eta_{R \cdot V(E')} \cdot f = f$, which follows trivially from the adjunction equations and U, R a cartesian reflection, since $\varepsilon = Id$ implies $\eta_{R(C)} = Id_{R(C)}$.

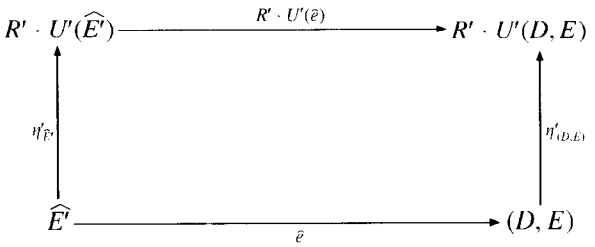
By construction we have $U' \cdot R' = Id_{\mathbf{E}}$, and ε'_E is given by the adjunction equations as

$$(Id_{R \cdot V(E)}, Id_E) = Id_{R'(E)} = R'(\varepsilon'_E) \cdot \eta'_{R'(E)}$$

$$= (R \cdot V(\varepsilon'_E), \varepsilon'_E) \cdot (\eta_{R \cdot V(E)}, Id_E) = (R \cdot V(\varepsilon'_E) \cdot \eta_{R \cdot V(E)}, \varepsilon'_E),$$

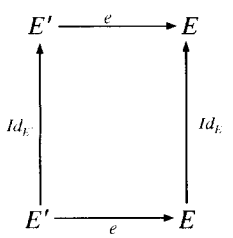
i.e., $\varepsilon'_E = Id_E$.

To finish proving the proposition, we now have to exhibit for each $(D, E) \in |\mathbf{D} \underset{\mathbf{C}}{\times} \mathbf{E}_1|$ and each $e : E' \to E$, where of course $E = U'(D, E)$, an arrow $\hat{e} : \widehat{E'} \to (D, E)$ in $\mathbf{D} \underset{\mathbf{C}}{\times} \mathbf{E}$ such that $U'(\hat{e}) = e$ and the square

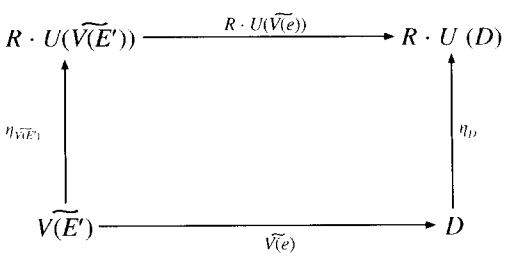


is a pullback.

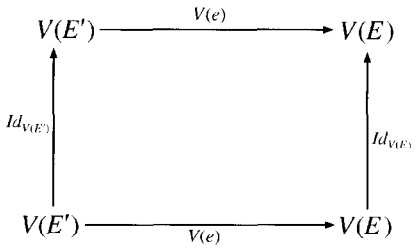
Choose $\hat{e} = (\widetilde{V(e)}, e) : (\widetilde{V(E')}, E') \to D, E$ where $\widetilde{V(e)} : \widetilde{V(E')} \to D$ in $\mathbf{D}$ is the lifting of $V(e) : V(E') \to U(D)$ provided by the cartesian reflection U, R. Obviously, $U'(\hat{e}) = e$. To check that for such a map $\hat{e}$ the above square is a pullback, notice the equality $R' \cdot U'(\hat{e}) = (R \cdot V(e), e) = (R \cdot U(\widetilde{V(e)}), e)$. Therefore, the projection functors U' and V' send the above natural square to the respective diagrams



and



in **E** and **D**, which are then both mapped by V and U to the diagram



Now, notice that these last three squares are all pullback squares, since two are pullbacks of identities, and the other is a pullback by hypothesis. The proposition then follows from the following easy lemma, which is left to the reader.

Lemma. *Consider a pullback of functors like in figure* (‡) *but with no assumptions whatsoever about U and V. For any commuting square S in* $\mathbf{D} \underset{\mathbf{C}}{\times} \mathbf{E}$ *such that* $U'(S), V'(S)$, *and* $V \cdot U'(S)$ *are pullbacks in* **E**, **D**, *and* **C**, *then S is a pullback in* $\mathbf{D} \underset{\mathbf{C}}{\times} \mathbf{E}$. □

4. Borrowing logics

We are now ready to apply the extension and fibration results of Section 3 to the borrowing of logical structure. Specifically, we verify that for each of the seven forgetful functors in Section 2 the conditions in Corollary 22 apply, so that all of them are cartesian reflections and therefore fibrations. As a consequence, the extension map used to borrow the missing logical structure is a cartesian lifting.

In each case, the pullback construction yielding the extension map provides an explicit description of the new logical structure being obtained. We discuss such descriptions and prove several results showing that the borrowed structure enjoys good logical properties, either in general or under natural restrictions. We also illustrate the constructions with relevant examples.

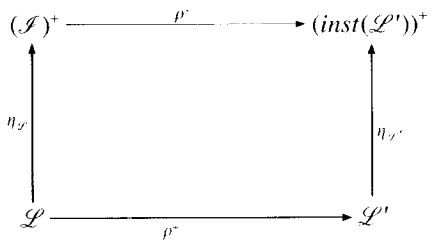
4.1. Endowing an institution with an entailment system

Since there are different ways of translating institutions (and hence logics), there are also several possibilities for building a logic on top of an institution $\mathscr{I}$ using an already known logic $\mathscr{L}'$ and a translation of $\mathscr{I}$ into the institution underlying $\mathscr{L}'$, by applying the construction introduced in the above section. The most promising case is that in which *maps of institutions* [24] are used, because completeness is preserved, and the entailment system $\vdash$ defined for $\mathscr{I}$ by the pullback can be easily described as the coding of the theories via a map and the application of the entailment system $\vdash'$ of $\mathscr{L}'$, so that any (finitary) description of $\vdash'$ is also a description of $\vdash$.

Although in general $\underline{Log}$ does not have pullbacks, they do exist in the particular case of pulling back the unit of the adjunction along the image under $(-)^+$ of a map of institutions.

Proposition 25. *The functors* $inst : \underline{Log} \to \underline{Inst}$ *and* $(-)^+ : \underline{Inst} \to \underline{Log}$ *form a cartesian reflection and therefore* $inst$ *is a fibration.*

Proof. By Proposition 7, $(-)^+$ is the right adjoint of $inst$, the counit of this adjunction is the identity natural transformation and the unit of this adjunction for a logic $\mathscr{L}$ is the embedding $\eta_{\mathscr{L}}$ of $\mathscr{L}$ into $(inst(\mathscr{L}))^+$. Thus, in order to show that these functors form a cartesian reflection, it suffices to show that for each institution $\mathscr{I} = (\mathbf{Sign}, Sen, Mod, \models)$, each logic $\mathscr{L}'$ and each map of institutions $\rho = (\Phi, \alpha, \beta) : \mathscr{I} \to inst(\mathscr{L}')$ the following diagram is a pullback, where $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$ and $\vdash$ is defined by $\Gamma \vdash_\Sigma \phi$ iff both $\alpha_\Sigma(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_\Sigma(\phi)$ and $\Gamma \vdash_{\models_\Sigma} \phi$.



First of all note that $\mathscr{L}$ is a logic, because, by definition of $\vdash$, if $\Gamma \vdash_\Sigma \phi$, then $\Gamma \vdash_{\models_\Sigma} \phi$; moreover, by definition of $\mathscr{L}$, $inst(\mathscr{L}) = \mathscr{I}$ and hence $\eta_{\mathscr{L}} : \mathscr{L} \to (\mathscr{I})^+$ is a unit map. The definition of $\vdash$ guarantees also that ρ^- is a map from $\mathscr{L}$ into $\mathscr{L}'$, because if $\Gamma \vdash_\Sigma \phi$, then – by the very definition of $\vdash$ – we have $\alpha_\Sigma(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_\Sigma(\phi)$. Obviously the diagram commutes, so we only have to check the universal property.

Let us assume that $\rho^+ \cdot (\Phi_1, \alpha_1, \beta_1) = \eta_{\mathscr{L}'} \cdot (\Phi_2, \alpha_2, \beta_2)$, for some logic $\mathscr{L}''$ and $(\Phi_1, \alpha_1, \beta_1) : \mathscr{L}'' \to (\mathscr{I})^+$, $(\Phi_2, \alpha_2, \beta_2) : \mathscr{L}'' \to \mathscr{L}'$.

Since the components of $\eta_{\mathscr{L}'}$ are all identitites, $\rho^+ \cdot (\Phi_1, \alpha_1, \beta_1) = \eta_{\mathscr{L}'} \cdot (\Phi_2, \alpha_2, \beta_2)$ implies that $\Phi_2 = \Phi \cdot \Phi_1$, $\alpha_2 = \alpha_{\Phi_1} \circ \alpha_1$ and $\beta_2 = \beta_1 \circ \beta_{\Phi_1}$.

Let us show that $(\Phi_1, \alpha_1, \beta_1)$ is a map of institutions from $\mathscr{L}''$ to $\mathscr{L}$; for this it is sufficient to show that $\Gamma \vdash''_\Sigma \phi$, denoting $\Phi_1(\Sigma, \emptyset)$ by (Σ', Γ'), implies $\alpha_{1\Sigma}(\Gamma) \cup \Gamma' \vdash_{\Sigma'} \alpha_{1\Sigma}(\phi)$, i.e. both $\alpha_{\Sigma'}(\alpha_{1\Sigma}(\Gamma) \cup \Gamma') \vdash'_{\Phi(\Sigma',\emptyset)} \alpha_{\Sigma'}(\alpha_{1\Sigma}(\phi))$ and $\alpha_{1\Sigma}(\Gamma) \cup \Gamma' \vdash_{\models_{\Sigma'}} \alpha_{1\Sigma}(\phi)$.

Since $\Phi_2 = \Phi \cdot \Phi_1$ and $\alpha_2 = \alpha_{\Phi_1} \circ \alpha_1$, the first condition holds, because $(\Phi_2, \alpha_2, \beta_2)$ is a map and hence preserves the entailment; moreover the second condition follows from the satisfaction condition for $(\Phi_1, \alpha_1, \beta_1)$ and the soundness of $\mathscr{L}''$.

Because $\eta_{\mathscr{L}}$ consists of identities, $(\Phi_1, \alpha_1, \beta_1)$ is the unique factorization required, and therefore the above diagram is a pullback, so that $inst$ and $(-)^+$ form a cartesian reflection. Therefore, by Corollary 22, $inst$ is a fibration. □

It is worth pointing out that completeness is preserved by this construction, as the following proposition shows.

Proposition 26. *Using the notation of Proposition* 25, *if* $\mathscr{L}'$ *is complete, then so is* $\mathscr{L}$.

Proof. Note that the logic $\mathscr{L}' = (\mathbf{Sign}', Sen', Mod', \models', \vdash')$ is complete iff $\vdash' = \vdash_{\models'}$. Therefore, $\mathscr{L}'$ is complete iff $\eta_{\mathscr{L}'} : \mathscr{L}' \to (inst(\mathscr{L}'))^+$ is an isomorphism. Since the pullback of an isomorphism is always an isomorphism, we then have $\eta_{\mathscr{L}} : \mathscr{L} \to (\mathscr{I})^+$ an isomorphism, and therefore $\mathscr{L}$ is complete. □

For any map ρ between institutions, this construction builds the entailment system that in [1] was denoted by $\vdash^{\rho}_{th}$ for $th = ax(\Phi(\Sigma))$.

For each institution $\mathscr{I} = (\mathbf{Sign}, Sen, Mod, \models)$, each logic $\mathscr{L}'$ and each map of institutions $\rho : \mathscr{I} \to inst(\mathscr{L}')$, with $\rho = (\Phi, \alpha, \beta)$, the extension of $\mathscr{I}$ by $(-)^+$ and *inst* is the logic $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$, that is the original institution $\mathscr{I}$ enriched by the entailment relation $\vdash$, defined by $\Gamma \vdash_{\Sigma} \phi$ iff both $\Gamma \vdash_{\models_{\Sigma}} \phi$ and $\alpha_{\Sigma}(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_{\Sigma}(\phi)$. Thus, the entailment system built for $\mathscr{I}$ can be informally described as obtained by:
- coding sentences by means of the map, that is, considering $\alpha_{\Sigma}(\Gamma)$ and $\alpha_{\Sigma}(\phi)$;
- using the entailment system of $\mathscr{L}'$, that is verifying whether $\alpha_{\Sigma}(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_{\Sigma}(\phi)$;
- checking the soundness of the deduction, that is, verifying whether $\Gamma \vdash_{\models_{\Sigma}} \phi$.

The last step is unnecessary if for each Σ-model there exists a $\Phi(\Sigma,\emptyset)$-model satisfying the same sentences (under translation), because in that case the soundness of $\vdash'$ w.r.t. $\models'$ guarantees that $\alpha_{\Sigma}(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_{\Sigma}(\phi)$ implies $\Gamma \vdash_{\models_{\Sigma}} \phi$.

Lemma 27. *Given an institution* $\mathscr{I} = (\mathbf{Sign}, Sen, Mod, \models)$, *a logic* $\mathscr{L}' = (\mathbf{Sign}', Sen', Mod', \models', \vdash')$ *and a map of institutions* $\rho : \mathscr{I} \to inst(\mathscr{L}')$, *with* $\rho = (\Phi, \alpha, \beta)$, *for each signature* Σ *in* **Sign** *if one of the following conditions holds, then* $\alpha_{\Sigma}(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_{\Sigma}(\phi)$ *implies* $\Gamma \vdash_{\models_{\Sigma}} \phi$ *for each* $\Gamma \cup \{\phi\} \subseteq Sen(\Sigma)$.

(i) *for each* $M \in |Mod(\Sigma)|$ *there is an* $M' \in |Mod'(\Phi(\Sigma,\emptyset))|$ *s.t.* $M \models_{\Sigma} \psi$ *iff* $M' \models_{sign(\Phi(\Sigma,\emptyset))} \alpha_{\Sigma}(\psi)$.

(ii) *for each* $M \in |Mod(\Sigma)|$ *there is an* $M' \in |Mod'(\Phi(\Sigma,\emptyset))|$ *s.t.* $\beta(M') = M$.

(iii) *any two isomorphic models in* $Mod(\Sigma)$ *satisfy the same sentences, and for each* $M \in |Mod(\Sigma)|$ *there is an* $M' \in |Mod'(\Phi(\Sigma,\emptyset))|$ *s.t.* $\beta(M')$ *is isomorphic to* M.

Proof. (i) Let us assume that $\alpha_{\Sigma}(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_{\Sigma}(\phi)$, for some $\Gamma \subseteq Sen(\Sigma)$ and $\phi \in Sen(\Sigma)$, and that $M \models_{\Sigma} \gamma$ for all $\gamma \in \Gamma$. Then, there is an $M' \in |Mod'(\Phi(\Sigma,\emptyset))|$ s.t. $M \models_{\Sigma} \psi$ iff $M' \models'_{sign(\Phi(\Sigma,\emptyset))} \alpha_{\Sigma}(\psi)$ and hence $M' \models'_{sign(\Phi(\Sigma,\emptyset))} \alpha_{\Sigma}(\gamma)$ for all $\gamma \in \Gamma$. Thus, by the soundness of $\vdash'$ w.r.t. $\models'$, $M' \models'_{sign(\Phi(\Sigma,\emptyset))} \alpha_{\Sigma}(\phi)$, and therefore $M \models_{\Sigma} \phi$.

(ii) Let us show that the previous condition (i) is satisfied. For each $M \in |Mod(\Sigma)|$ there is an $M' \in |Mod'(\Phi(\Sigma,\emptyset))|$ s.t. $\beta(M') = M$; then for all sentences $\psi \in Sen(\Sigma)$, since ρ is a map of institutions, $M = \beta(M') \models_{\Sigma} \psi$ iff $M' \models'_{sign(\Phi(\Sigma,\emptyset))} \alpha_{\Sigma}(\psi)$.

(iii) Let us again show that the previous condition (i) is satisfied. For each $M \in |Mod(\Sigma)|$ there is an $M' \in |Mod'(\Phi(\Sigma,\emptyset))|$ s.t. $\beta(M')$ is isomorphic to M; then, for all

sentences $\psi \in Sen(\Sigma)$, by ρ being a map of institutions we have, $M' \models'_{sign(\Phi(\Sigma,\emptyset))} \alpha_\Sigma(\psi)$ iff $\beta(M') \models_\Sigma \psi$ iff $M \models_\Sigma \psi$, because we have assumed that satisfaction is invariant under isomorphisms. □

Corollary 28. *Under the assumption that ρ satisfies one of the conditions in Lemma 27 we have*

$$\Gamma \vdash_\Sigma \phi \Leftrightarrow \alpha_\Sigma(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha_\Sigma(\phi).$$

Therefore, the lifted map $\rho^+ : \mathscr{L} \to \mathscr{L}'$ is a conservative *map of the underlying entailment systems.*

As a consequence, if ρ satisfies one of the conditions of Lemma 27 and its signature and sentence components are effective, and therefore Γ' in $\Phi(\Sigma,\Gamma) = (\Sigma',\Gamma')$ has an effective description if Γ has one, then an effective way of generating the entailment system of $\mathscr{L}'$ can also be used to effectively determine whether $\Gamma \vdash \phi$.

Example 29. Let us consider a quite classical example, the reduction of many-sorted conditional equational logic, denoted from now on $\mathscr{MS}$, to unsorted Horn-clause logic with equality, denoted from now on $\mathscr{UH}$, making explicit the typing of the variables (see e.g. [27]). In this example, as in the following ones, the notation for many-sorted open formulas has been slightly changed w.r.t. the usual algebraic notation, according to [13]. Indeed in order to make the translation of formulas along signature morphisms easier, a partial function $V : X \to S$, the *typing of variables*, is prefixed to any conditional formula on the signature (S,F) and variables $\{X_s\}_{s\in S}$, where $X_s = V^{-1}(s)$ are the s-typed variables. From now on we will assume that for each formula $V.\phi$ the domain of V is finite.

Let us informally sketch the elements of the map of institutions. Let us fix a many-sorted signature Σ with sorts S and function symbols F. The information about the typing of function symbols, that in the many-sorted framework is part of the signature, is now explicitly given by a set of one-sorted sentences, called *well-formedness axioms*; thus we define the translation of Σ into a one-sorted theory $\Phi(\Sigma)$, by setting $\Phi(\Sigma) = ((Op',P'),Ax')$, where

- Op'_n is the disjoint union of the $F_{s_1\ldots s_n,s}$, i.e., of the n-ary function symbol sets, disregarding type of arguments and results (so that any Σ-term is an (Op',P')-term, too);
- P' contains only the typing predicates, i.e. $P'_1 = \{- : s \mid s \in S\}$, where the symbol $-$ denotes the place of the argument in a postfix notation, and $P'_k = \emptyset\ \forall k \neq 1$; and
- Ax' consists of the sentences

$$x_1 : s_1 \wedge \cdots \wedge x_k : s_k \supset op(x_1,\ldots,x_k) : s$$

for each $op \in F_{s_1\ldots s_k,s}$.

With the help of the typing predicates, any many-sorted conditional equation over Σ can be translated into a one-sorted equivalent one over the signature of $\Phi(\Sigma)$; indeed

let us consider a many-sorted formula

$$\xi = V.(t_1 = t_1' \wedge \cdots \wedge t_n = t_n' \supset t = t')$$

over Σ and the variables x_i, where $V(x_i) = s_i$ for $i = 1 \ldots k$, and define

$$\alpha_\Sigma(\xi) = (x_1 : s_1 \wedge \cdots \wedge x_k : s_k \wedge t_1 = t_1' \wedge \cdots \wedge t_n = t_n' \supset t = t').$$

Then in $\alpha_\Sigma(\xi)$ the information about the typing of the variables is carried by the predicates $x_i : s_i$ in the premises.

Finally, any unsorted model A' of $\Phi(\Sigma)$ is mapped by β_Σ into the many-sorted algebra $A = (\{s^A\}_{s \in S}, \{f^A\}_{f \in F})$, where $s^A = \{a \in A' \mid a : s^{A'}\}$ and f^A is the restriction of $f^{A'}$ to $s_1^A \times \cdots \times s_n^A$. Note that, since A' satisfies the well-formedness axioms whenever the arguments of a function are appropriately typed, the result is also appropriately typed, i.e. $a_i : s_i^{A'}$ for $i = 1 \ldots n$ implies $f^{A'}(a_1, \ldots, a_n) : s^{A'}$ for each $f \in F_{s_1 \ldots s_n, s}$, and hence the interpretation of the function symbols in A yields total functions. It is easy to check that A' satisfies $\alpha_\Sigma(\xi)$ iff A satisfies ξ.

It is also worth noting that β satisfies condition (iii) of Lemma 27. Indeed satisfaction in $\mathscr{MS}$ is obviously invariant under isomorphism and for each many-sorted algebra A on a signature Σ the following one-sorted algebra A' is a model of $\Phi(\Sigma)$ and its image along β_Σ is isomorphic to A:

- the carrier of A' is the disjoint union[8] of the carriers s^A for all $s \in S$ and of a special element $\{\perp\}$;
- $f^{A'}(a_1, \ldots, a_n) = f^A(a_1, \ldots, a_n)$ if $f \in F_{s_1 \ldots s_n, s}$ and $a_i \in s_i^A$; otherwise $f^{A'}(a_1, \ldots, a_n) = \perp$;
- $a : s$ iff $a \in s^A$ for all sorts s.

Let us now endow the unsorted Horn-clause logic with equality institution with an entailment system to build a logic $\mathscr{L}_{\mathscr{HH}}$, choosing the entailment generated by (a version[9] of) the classical Birkhoff's deductive system.

For any one-sorted signature $\Sigma = (Op, P)$ and any set Γ of Horn clauses with equality on $\Sigma, \Gamma \vdash_{B\Sigma} \phi$ iff ϕ is in the inductive closure of Γ and the axioms qualifying the equality, where the possibly decorated x's are pairwise different variables:

$$x = x$$

$$x = x' \supset x' = x$$

$$x = x' \wedge x' = x'' \supset x = x''$$

$$x_1 = x_1' \wedge \cdots \wedge x_n = x_n' \wedge p(x_1, \ldots, x_n) \supset p(x_1', \ldots, x_n')$$

$$x_1 = x_1' \wedge \cdots \wedge x_n = x_n' \supset f(x_1, \ldots, x_n) = f(x_1', \ldots, x_n')$$

[8] With a slight abuse of notation we use below the same symbol a for an element $a \in s^A$ and for its disjoint copy.

[9] The weakening rule is usually not included in the definition of the classical Birkhoff system, but it is needed to achieve a system complete w.r.t. *conditional* sentences.

w.r.t. the following inference rules of weakening, instantiation and modus ponens.

$$\textit{weakening}\ \frac{\varepsilon_1 \wedge \cdots \wedge \varepsilon_n \supset \varepsilon}{\varepsilon_1 \wedge \cdots \wedge \varepsilon_n \wedge \eta_1 \wedge \cdots \wedge \eta_k \supset \varepsilon}$$

$$\textit{instantiation}\ \frac{\phi}{V(\phi)}\quad V \text{ homomorphism of the term algebras}$$

$$\textit{modus ponens}\ \frac{\varepsilon_1 \wedge \cdots \wedge \varepsilon_n \supset \varepsilon, \quad \eta_1 \wedge \cdots \wedge \eta_k \supset \varepsilon_i}{\varepsilon_1 \wedge \cdots \wedge \varepsilon_{i-1} \wedge \eta_1 \wedge \cdots \wedge \eta_k \wedge \varepsilon_{i+1} \wedge \cdots \wedge \varepsilon_n \supset \varepsilon}$$

Since it is defined by inductive closure, $\vdash_B$ is monotonic, transitive and reflexive; moreover it is straightforward to check from the definition that it also satisfies the $\vdash$-translation condition and that it is sound, so that $\mathscr{L}_{\#\#} = (\mathbf{Sign}_{\#\#}, Sen_{\#\#}, Mod_{\#\#}, \vdash_B, \models_{\#\#})$ is a logic.

Then, applying Lemma 27 and Corollary 28, we can borrow $\vdash_B$ to build an entailment system $\vdash$ for $\mathscr{MS}$, defined by:

for any many-sorted signature $\Sigma = (S, F)$ and any set Γ of conditional sentences on Σ,

$$\Gamma \vdash_\Sigma V.(t_1 = t_1' \wedge \cdots \wedge t_n = t_n' \supset t = t')$$

(say, with $V(x_i) = s_i$ for $i = 1 \ldots k$, and $V(x)$ undefined otherwise) iff

$$x_1 : s_1 \wedge \cdots \wedge x_k : s_k \wedge t_1 = t_1' \wedge \cdots \wedge t_n = t_n' \supset t = t'$$

is in the inductive closure of the following axioms, where the possibly decorated x's are pairwise different variables:

Well-formedness

$$x_1 : s_1 \wedge \cdots \wedge x_k : s_k \supset op(x_1, \ldots, x_k) : s$$

for all $op \in F_{s_1 \ldots s_k, s}$.

Proper axioms

$$x_1 : s_1 \wedge \cdots \wedge x_k : s_k \wedge t_1 = t_1' \wedge \cdots \wedge t_n = t_n' \supset t = t'$$

for all $(V.t_1 = t_1' \wedge \cdots \wedge t_n = t_n' \supset t = t') \in \Gamma$ with $V = \{(x_i, s_i) \mid i = 1 \ldots k\}$.

Equality axioms

$$x = x$$

$$x = x' \supset x' = x$$

$$x = x' \wedge x' = x'' \supset x = x''$$

$$x = x' \wedge x : s \supset x' : s$$

$$x_1 = x_1' \wedge \cdots \wedge x_n = x_n' \supset f(x_1, \ldots, x_n) = f(x_1', \ldots, x_n')$$

w.r.t. the inference rules of weakening, instantiation and modus ponens.

Note that all the deductions made by this system are sound, because of Lemma 27.

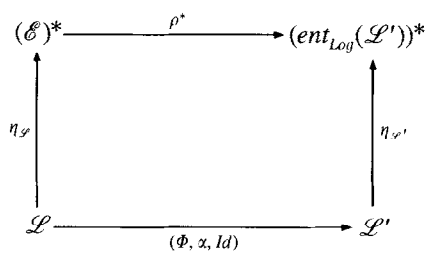
Since $\vdash_B$ is complete w.r.t. conditional sentences, as it is provable by standard techniques, this completeness is inherited by $\vdash$, because of Proposition 26; thus we have a complete logic $\mathscr{L}_{\mathscr{MS}} = (\mathbf{Sign}_{\mathscr{MS}}, Sen_{\mathscr{MS}}, Mod_{\mathscr{MS}}, \vdash, \models_{\mathscr{MS}})$ for the many-sorted institution.

4.2. *Endowing an entailment system with an institution*

Sometimes a logic is developed before a general agreement on its model theory is reached. We show that a borrowing construction can then be applied to provide the missing model theory, and that completeness is preserved if the map used is conservative. We illustrate this construction with the borrowing of algebraic models for linear logic.

Proposition 30. *The functors* $ent_{Log} : \underline{Log} \to \underline{Ent}$ *and* $(-)^* : \underline{Ent} \to \underline{Log}$ *form a cartesian reflection and therefore* ent_{Log} *is a fibration.*

Proof. By Proposition 9, $(-)^*$ is the right adjoint of ent_{Log}, the counit of this adjunction is the identity natural transformation, and the unit of this adjunction for a logic $\mathscr{L}$ is the embedding $\eta_{\mathscr{L}}$ of $\mathscr{L}$ into $(ent_{Log}(\mathscr{L}))^*$. Thus, in order to prove that ent_{Log} and $(-)^*$ form a cartesian reflection, it suffices to show that for each entailment system $\mathscr{E} = (\mathbf{Sign}, Sen, \vdash)$, each logic $\mathscr{L}' = (\mathbf{Sign}', Sen', Mod', \models', \vdash')$ and each map of entailment systems $\rho = (\Phi, \alpha) : \mathscr{E} \to ent_{Log}(\mathscr{L}')$ the following diagram is a pullback, where $\mathscr{L} = (\mathbf{Sign}, Sen, Mod, \models, \vdash)$, $Mod = Mod' \cdot \Phi^{op}$, and $M \models_\Sigma \phi$ if and only if $M \models'_{sign(\Phi(\Sigma,\emptyset))} \alpha_\Sigma(\phi)$.



First of all note that $\mathscr{L}$ is a logic, because Mod is a functor, as it is the composition of functors, the satisfaction condition is satisfied, because it is satisfied by $\models'$ and if $\Gamma \vdash \phi$, then $\alpha(\Gamma) \vdash' \alpha(\phi)$ and hence, by soundness of $\mathscr{L}'$, $\alpha(\Gamma) \vdash_{\models'} \alpha(\phi)$, so that $\Gamma \vdash_{\models} \phi$.

Moreover, by definition of $\mathscr{L}$, $ent_{Log}(\mathscr{L}) = \mathscr{E}$ and hence $\eta_{\mathscr{L}} : \mathscr{L} \to (\mathscr{E})^*$ is a unit map of logics and (Φ, α) is a map of entailment systems from $ent_{Log}(\mathscr{L})$ into $ent_{Log}(\mathscr{L}')$; by definition of Mod and $\models$ the identity is a natural transformation from $Mod' \cdot \Phi^{op}$ to $Mod = Mod' \cdot \Phi^{op}$ verifying the satisfaction condition, so that (Φ, α, Id) is a map of logics from $\mathscr{L}$ into $\mathscr{L}'$. Obviously the diagram commutes, so we only have to check the universal property.

Let us assume that $\rho^* \cdot (\Phi_1, \alpha_1, \beta_1) = \eta_{\mathscr{L}'} \cdot (\Phi_2, \alpha_2, \beta_2)$, for some logic $\mathscr{L}''$ and $(\Phi_1, \alpha_1, \beta_1) : \mathscr{L}'' \to (\mathscr{E})^*$, $(\Phi_2, \alpha_2, \beta_2) : \mathscr{L}'' \to \mathscr{L}'$, i.e. that $\Phi \cdot \Phi_1 = Id_{\mathbf{Sign}} \cdot \Phi_2 = \Phi_2$, $\alpha_{\Phi_1} \circ \alpha_1 = Id_{Sen\Phi_2} \circ \alpha_2 = \alpha_2$ and $\beta_1 \circ Id_{\emptyset\Phi_1} = \beta_2 \circ \beta_{\emptyset\Phi_2}$.

Since $(\Phi_1, \alpha_1, \beta_1) : \mathscr{L}'' \to (\mathscr{E})^*$, $ent_{Log}(\Phi_1, \alpha_1, \beta_1) : ent_{Log}(\mathscr{L}'') \to ent_{Log}((\mathscr{E})^*)$, i.e. (Φ_1, α_1) is a map of entailment systems from $ent_{Log}(\mathscr{L}'')$ into $\mathscr{E} = ent_{Log}(\mathscr{L})$. And β_2 is a natural transformation from $Mod' \cdot \Phi_2^{op} = Mod' \cdot \Phi^{op} \cdot \Phi_1^{op} = Mod \cdot \Phi_1^{op}$ to Mod'', and for all signatures Σ in $\mathbf{Sign}''$, all sentences $\phi \in Sen''(\Sigma)$ and all models $M \in |Mod(\Phi_1(\Sigma, \emptyset))|$ we have $M \models_{sign(\Phi_1(\Sigma,\emptyset))} \alpha_1(\phi)$ if and only if $M \models'_{sign(\Phi(\Phi_1(\Sigma,\emptyset)))} \alpha_{sign(\Phi_1(\Sigma,\emptyset))}(\alpha_{1\Sigma}(\phi))$, i.e. iff $M \models'_{sign(\Phi_2(\Sigma,\emptyset))} \alpha_{2\Sigma}(\phi)$; since $(\Phi_2, \alpha_2, \beta_2)$ is a map of logics, $M \models'_{sign(\Phi_2(\Sigma,\emptyset))} \alpha_2(\phi)$ iff $\beta_2(M) \models''_{\Sigma} \phi$. Therefore $(\Phi_1, \alpha_1, \beta_2)$ is a map of logics from $\mathscr{L}''$ into $\mathscr{L}$ and hence it is the unique factorization required.

So the above diagram is a pullback; hence ent_{Log} and $(-)^*$ form a cartesian reflection. Therefore, by Corollary 22, ent_{Log} is a fibration. □

Proposition 31. *Given an entailment system, $\mathscr{E}$, a logic $\mathscr{L}'$ and a conservative map of entailment systems, $(\Phi, \alpha) : \mathscr{E} \to ent_{Log}(\mathscr{L}')$, let $(\Phi, \alpha, Id) : \mathscr{L} \to \mathscr{L}'$ be the cartesian lifting of (Φ, α) to a map of logics constructed in Proposition* 30.

If $\mathscr{L}'$ is complete, then so is $\mathscr{L}$.

Proof. We need to show that ($M \models_\Sigma \phi$ for all $M \in Mod(\Sigma, \Gamma)$) iff $\Gamma \vdash_\Sigma \phi$; but, by definition of $\models_\Sigma$, the first condition can be rewritten as $M' \models'_{\Sigma'} \alpha(\phi)$ for all $M' \in Mod'(\Sigma', \Gamma'')$, where $(\Sigma', \Gamma'') = \Phi(\Sigma, \Gamma)$ and $(\Sigma', \Gamma') = \Phi(\Sigma, \emptyset)$.

And, by completeness of $\mathscr{L}'$, we have that

$$M' \models'_{\Sigma'} \alpha(\phi) \quad \text{for all } M' \in Mod'(\Sigma', \Gamma'')$$

if and only if $\Gamma'' \vdash'_{\Sigma'} \alpha(\phi)$.

Since by Φ being α-sensible we have $thm'(\Sigma', \Gamma'') = thm'(\Sigma', \Gamma' \cup \alpha(\Gamma))$, we can rewrite the last equivalence as ($M' \models'_{\Sigma'} \alpha(\phi)$ for all $M' \in Mod'(\Sigma', \Gamma'')$) if and only if $\Gamma' \cup \alpha(\Gamma) \vdash'_{\Sigma'} \alpha(\phi)$, or, in more compact form, ($M' \models'_{\Sigma'} \alpha(\phi)$ for all $M' \in Mod'(\Sigma', \Gamma'')$) if and only if $\alpha(\Gamma) \vdash'_{\Phi(\Sigma,\emptyset)} \alpha(\phi)$, which by (Φ, α) being conservative yields ($M' \models'_{\Sigma'} \alpha(\phi)$ for all $M' \in Mod'(\Sigma', \Gamma'')$) if and only if $\Gamma \vdash_\Sigma \phi$, as desired. □

Example 32. The proof theory of linear logic [9] reached a mature stage before systematic approaches to its model theory were attempted. Girard himself has proposed quite different kinds of models such as coherent spaces [9], lattice-theoretic models such as his phase semantics [9], and Hilbert space models [10]. Although attempts to systematize the model theory of linear logic using category theory have been made (see for example [32, 22, 21]), particular models may not quite fit even those general notions of model. However, lattice-theoretic models, including quantales, are in a sense the simplest, and are sufficient for completeness arguments. In hindsight we can view the process of endowing linear logic with such lattice-theoretic models as a borrowing of models from conditional equational logic. We explain below the details of this borrowing process and its good properties.

Following [21] we call the lattice models *Girard algebras*. Exploiting results of Cockett and Seely on weakly distributive categories [4] we can define a Girard algebra G as a set, together with binary operations $\oplus$, $\&$, $\otimes$, $\parr$, unary operations $\_^{\perp}$ and $!\_$, and constants $0, \top, 1, \perp$ such that:

(i) $(G, \oplus, 0)$ and $(G, \&, \top)$ are commutative and idempotent monoid structures, and in addition satisfy the equations

$$(x \oplus y) \& x = x$$

$$(x \& y) \oplus x = x$$

that is, they make G into a lattice with top $\top$ and bottom 0. Note that an order relation is then equationally definable by

$$x \leqslant y \quad \text{iff } x \oplus y = y$$

(ii) $(G, \otimes, 1)$ and $(G, \parr, \perp)$ are commutative monoid structures satisfying the monotonicity conditional equations

$$(x \oplus y = y) \quad \text{and } (x' \oplus y' = y') \quad \text{implies } (x \otimes x') \oplus (y \otimes y') = (y \otimes y')$$

$$(x \oplus y = y) \quad \text{and } (x' \oplus y' = y') \quad \text{implies } (x \parr x') \oplus (y \parr y') = (y \parr y')$$

and the weak distributivity equation

$$(x \otimes (y \parr z)) \oplus ((x \otimes y) \parr z) = (x \otimes y) \parr z$$

(iii) $\_^{\perp}$ and $!\_$ are unary operations satisfying the equations

$$(x \otimes x^{\perp}) \oplus \perp = \perp$$

$$(x \parr x^{\perp}) \oplus 1 = x \parr x^{\perp}$$

$$(x \oplus y = y) \text{ implies } !x \oplus !y = !y$$

$$(!x) \oplus x = x$$

$$(!!x) \oplus !x = !!x$$

$$!x = (!x) \otimes (!x)$$

$$!\top = 1$$

(iv) If desired, one can define two more operations $\_\multimap\_$ and $?\_$ by the defining equations

$$x \multimap y = x^{\perp} \parr y$$

$$?x = (!(x^{\perp}))^{\perp}$$

Let (Σ_G, E_G) be the conditional equational theory specifying the class of Girard algebras just defined.

A *signature* in propositional linear logic is just a set S of propositional constants. The set of *sentences* of linear logic $Sen_{LL}(S)$ on such a signature is the set of sequents [10] of the form

$$A_1 \ldots A_n \vdash B_1 \ldots B_m$$

where $A_i, B_j \in T_{\Sigma_G}(S)$, are Σ_G-terms on the constants S, for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$. A map of signatures is just a function $f : S \to S'$. It induces a Σ_G-homomorphism $T_{\Sigma_G}(f) : T_{\Sigma_G}(S) \to T_{\Sigma_G}(S')$ and therefore an obvious function $Sen_{LL}(f) : Sen_{LL}(S) \to Sen_{LL}(S')$.

Let $\mathscr{LL}$ denote the entailment system of linear logic as defined by its rules of deduction in any of its formulations, and let $\mathscr{CE}$ denote (unsorted) conditional equational logic. We can define a map of entailment systems

$$(\Phi, \alpha) : \mathscr{LL} \to ent_{Log}(\mathscr{CE})$$

where $\Phi(S, \emptyset) = (\Sigma_G \cup S, E_G)$, α maps each S-sequent

$$A_1 \ldots A_n \vdash B_1 \ldots B_m$$

to the $\Sigma_G \cup S$-equation

$$(A_1 \otimes \cdots \otimes A_n) \oplus (B_1 \parr \cdots \parr B_m) = (B_1 \parr \cdots \parr B_m)$$

and in general

$$\Phi(S, \Delta) = (\Sigma_G \cup S,\ E_G \cup \alpha(\Delta)).$$

It can be shown using ideas in [21, 4] that the above map (Φ, α) is indeed conservative and therefore, by Proposition 31, that the borrowing of the conditional logic institution via (Φ, α) makes $\mathscr{LL}$ into a complete logic. This borrowing process amounts to a systematic reduction of propositional linear logic to algebra in a way analogous to the reductions of classical propositional logic to Boolean algebra and of intuitionistic propositional logic to Heyting algebra. In fact, these two latter classes of algebras correspond to equationally defined subclasses of Girard algebras.

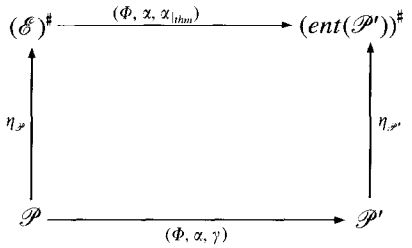
4.3. Endowing an entailment system with a proof calculus

Let us now apply the general results of Section 3 to the adjunction between entailment systems and proof calculi to show how an informative proof calculus can be added to any entailment system $\mathscr{E}$, provided we are given a proof calculus $\mathscr{P}$ and a translation of $\mathscr{E}$ into the entailment system underlying $\mathscr{P}$.

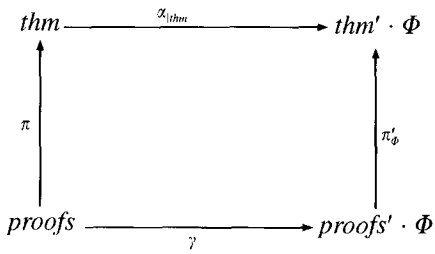
[10] As in [32, 22] we adopt a two-sided sequent presentation. However, we could have adopted instead a one-sided presentation as in [9].

Proposition 33. *The functors ent* : $\underline{PCalc} \to \underline{Ent}$ *and* $(-)^\sharp$: $\underline{Ent} \to \underline{PCalc}$ *form a cartesian reflection and therefore ent is a fibration.*

Proof. By Proposition 11, the composition $ent \cdot (-)^\sharp$ is the identity, the counit of the adjunction is the identity and the unit $\eta_{\mathscr{P}}$ for a proof calculus $\mathscr{P} = (\mathbf{Sign}, Sen, \vdash, P, Pr, \pi)$ is $(Id_{\mathbf{Sign}}, Id_{Sen}, \pi)$. Thus, in order to show that *ent* and $(-)^\sharp$ form a cartesian reflection, it suffices to prove that the following diagram is a pullback



where $\mathscr{P} = (\mathbf{Sign}, Sen, \vdash, P, Pr, \pi)$, with $P = proofs$ and $Pr = Id_{\mathbf{Set}}$, and where π, γ, and *proofs* are defined by the following pullback square in the functor category $\mathbf{Set}^{\mathbf{Th}_0}$.



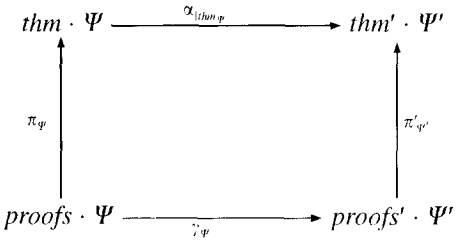
Since limits in functor categories are obtained by pointwise evaluation [18, Theorem 25.6] and **Set** has pullbacks, the pullback of the second square exists and is defined pointwise. Therefore, since every $\pi'_{\Phi(\Sigma,\Gamma)}$ is surjective, each $\pi_{(\Sigma,\Gamma)}$ is also surjective.[11] Thus $\mathscr{P}$ is a proof calculus.

Moreover, by definition (Φ, α, γ) is a map from $\mathscr{P}$ into $\mathscr{P}'$, because $\alpha \circ \pi = \pi'_\Phi \circ \gamma$, and $(Id_{\mathbf{Sign}}, Id_{Sen}, \pi)$ is a map from $\mathscr{P}$ into $(\mathscr{E})^\sharp$, because $Id_{Sen} \circ \pi = j_{Id_{\mathbf{Sign}}} \circ \pi$. And obviously the first diagram commutes; thus we only have to show that any other pair of maps of proof calculi that makes the diagram commute factorizes in a unique way through $\mathscr{P}$.

Let us assume that $(\Phi, \alpha, \alpha_{|thm}) \cdot (\Psi, \bar{\alpha}, \bar{\gamma}) = (Id_{\mathbf{Sign}'}, Id_{Sen'}, \pi') \cdot (\Psi, \bar{\alpha}', \bar{\gamma}')$, for some $\mathscr{P}''$ and $(\Psi, \bar{\alpha}, \bar{\gamma}) : \mathscr{P}'' \to (\mathscr{E})^\sharp, (\Psi, \bar{\alpha}', \bar{\gamma}') : \mathscr{P}'' \to \mathscr{P}'$.

[11] The property that the pullback of a surjective map in **Set** is also surjective is easy to check directly, and follows also (by the axiom of choice) from the general fact that in any category pullbacks of retracts are retracts, see e.g. [18, Proposition 21.13].

Since pullbacks in functor categories are computed pointwise [18, Theorem 25.6], the following diagram is pullback, because it is the translation of the diagram defining *proofs*, γ, π along Ψ (recalling that $\Phi \cdot \Psi = \Psi'$):



and hence, since $\alpha|_{thm_\Psi} \circ \overline{\gamma} = \pi'_{\Psi'} \circ \overline{\gamma}'$, there is a unique $\gamma'' : proofs'' \Rightarrow proofs \cdot \Psi$ s.t. both $\pi_\Psi \circ \gamma'' = \overline{\gamma}$ and $\gamma_\Psi \circ \gamma'' = \overline{\gamma}'$. Thus we only have to show that $(\Psi, \overline{\alpha}, \gamma'')$ is a map, i.e. that $\pi_\Psi \circ \gamma'' = \overline{\alpha} \circ \pi''$.

Since $(\Psi, \overline{\alpha}, \overline{\gamma})$ is a map, $j_\Psi \circ \overline{\gamma} = \overline{\alpha} \circ \pi''$, i.e., j being the identity on *thm*, $\overline{\gamma} = \overline{\alpha} \circ \pi''$; moreover, by definition of $\gamma'', \overline{\gamma} = \pi_\Psi \circ \gamma''$, so that $\pi_\Psi \circ \gamma'' = \overline{\alpha} \circ \pi''$.

Therefore *ent* and $(\_)^\natural$ form a cartesian reflection, and hence, by Corollary 22, *ent* is a fibration. □

Note that, according to the pullback construction in Proposition 33, the proof calculus $\mathscr{P}$ borrowed from $\mathscr{P}'$ via the map of entailment systems (Φ, α) from $\mathscr{E}$ in $ent(\mathscr{P}')$ associates to each theory T the set of proofs

$$proofs(T) = \{(\phi, p) \mid \phi \in thm(T) \text{ and } p \in proofs'(\Phi(T)) \text{ and } \alpha(\phi) = \pi'(p)\}.$$

Therefore a proof in $\mathscr{P}$ of a sentence ϕ consists of:

(i) checking that ϕ is a theorem of T, i.e. that $\phi \in thm(T)$;

(ii) a proof $p \in proofs'(\Phi(T))$ of $\alpha(\phi)$.

Of course, the proof calculus $\mathscr{P}$ is quite unsatisfactory, since in addition to having a proof of $\alpha(\phi)$ we have also to check that ϕ was a theorem of T in the first place. The desirable situation would be one in which the check in (i) becomes unnecessary sine then, assuming that Φ and α are computationally effective and that $\mathscr{P}'$ is also computationally effective, we will obtain a computationally effective way to prove theorems for $\mathscr{E}$, namely by mapping a sentence ϕ to the sentence $\alpha(\phi)$ and searching for a proof of $\alpha(\phi)$ in $\mathscr{P}'$. That is, we would like a set of proofs of the form

$$proofs(T) = \{(\phi, p) \mid \phi \in Sen(T) \text{ and } p \in proofs'(\Phi(T)) \text{ and } \alpha(\phi) = \pi'(p)\}.$$

This indeed is the case when (Φ, α) is a *conservative* map of entailment systems.

Lemma 34. *Let* $\mathscr{E} = (\mathbf{Sign}, Sen, \vdash)$ *be an entailment system, let* $\mathscr{P}' = (\mathbf{Sign}', Sen', \vdash', P', Pr', \pi')$ *be a proof calculus, and let* $(\Phi, \alpha) : \mathscr{E} \to ent(\mathscr{P}')$ *be a conservative map of*

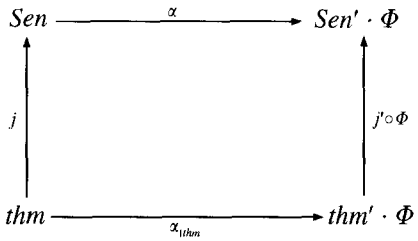
entailment systems. Then, for each theory $T \in \mathbf{Th}_0$, *the set* $proofs(T)$ *constructed in Proposition* 33 *for the proof calculus* $\mathscr{P}$ *borrowed from* $\mathscr{P}'$ *via* (Φ, α) *is the set*

$$proofs(T) = \{(\phi, p) \mid \phi \in Sen(T) \text{ and } p \in proofs'(\Phi(T)) \text{ and } \alpha(\phi) = \pi'(p)\}.$$

Proof. The key observation is that, using the fact that Φ is α-sensible, (Φ, α) is conservative if and only if for each $T \in \mathbf{Th}_0$

$$\phi \in thm(T) \Leftrightarrow \alpha(\phi) \in thm'(\Phi(T))$$

and this equivalence can be expressed in a compact way by saying that the diagram of natural transformations



is a pullback in $\mathbf{Set}^{\mathbf{Th}_0}$, where $j : thm \hookrightarrow Sen$ and $j' : thm' \hookrightarrow Sen'$ are the obvious subfunctors inclusions. The result then follows easily from the pointwise computation of pullbacks in $\mathbf{Set}^{\mathbf{Th}_0}$ by pasting the above diagram on top of the pullback diagram in the proof of Proposition 33, and noticing that the pasting of two such pullback diagrams is always a pullback diagram. □

Example 35. Let us consider again the example of the translation of many-sorted conditional equational logic into unsorted Horn logic with equality from Example 29.

We first endow the unsorted entailment system $ent_{Log}(\mathscr{L}_{\mathscr{U}\mathscr{H}})$ with a proof structure to get a proof calculus $\mathscr{PC}_{\mathscr{U}\mathscr{H}}$ and then, using the *conservative* (by Corollary 28) map of entailment systems $(\Phi, \alpha) : ent_{Log}(\mathscr{L}_{\mathscr{M}\mathscr{S}}) \to ent_{Log}(\mathscr{L}_{\mathscr{U}\mathscr{H}})$, sketched in Example 29, we build a many-sorted proof calculus $\mathscr{PC}_{\mathscr{M}\mathscr{S}}$, by applying Lemma 34.

Since the definition of the Birkhoff's entailment system is done by induction, we have at hand a natural structure for proofs in the unsorted framework. Let us indeed consider as proof structures just the proof trees. Thus, $\mathbf{Struct}_{\mathscr{U}\mathscr{H}}$ is the category of algebras on a tree signature (i.e. an algebraic signature including the sort *trees* and operations to define and deal with trees, like $root : trees \to sentences$)[12] and $P_{\mathscr{U}\mathscr{H}}$:

[12] Some details in the example are intentionally left unspecified, so that it can be realized with different possible choices of underlying algebraic framework; for instance, algebras could be simply many-sorted total algebras, as well as order-sorted (for example with the sentences seen also as trees with one node) or partial (allowing the constructor and selectors of proof trees to be partial); the signature could include the elements to describe sentences, and in this case the constructors of trees could be exactly the inference rules of the system, or could abstract from the particular kind of sentences and use standard constructors for trees and so on.

$Th_{0\mathscr{U}\mathscr{H}} \to \mathbf{Struct}_{\mathscr{U}\mathscr{H}}$ associates with any theory (Σ, Γ) the free algebra of trees whose nodes are labeled on $Sen_{\mathscr{U}\mathscr{H}}(\Sigma)$ and s.t. each leaf is in Γ or is an equality axiom, i.e. it is $t = t$, or $t = t' \supset t' = t$, or $t = t' \wedge t' = t'' \supset t = t''$, or $t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \supset f(t_1 \ldots, t_n) = f(t'_1, \ldots, t'_n)$, or $t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \wedge p(t_1, \ldots, t_n) \supset p(t'_1, \ldots, t'_n)$, and for each node there exists an inference rule among weakening, instantiation and modus ponens s.t. the (label of the) node is the consequence and (the labels of) its children are the premises of this rule. It is easy to check that, by the freeness of $P_{\mathscr{U}\mathscr{H}}(\Sigma, \Gamma)$, any homomorphism of theories in $Th_{0\mathscr{U}\mathscr{H}}$ induces a homomorphic translation of proof trees, so that $P_{\mathscr{U}\mathscr{H}}$ is a functor.

Moreover $Pr_{\mathscr{U}\mathscr{H}} : \mathbf{Struct}_{\mathscr{U}\mathscr{H}} \to \mathbf{Set}$ associates any algebra with its *tree* carrier and $\pi_{\mathscr{U}\mathscr{H}} : proofs_{\mathscr{U}\mathscr{H}} \Rightarrow Sen_{\mathscr{U}\mathscr{H}}$ sends any proof tree into its root. By definition of $\vdash_B$, $\mathscr{PC}_{\mathscr{U}\mathscr{H}} = (\mathbf{Sign}_{\mathscr{U}\mathscr{H}}, Sen_{\mathscr{U}\mathscr{H}}, \vdash_B, P_{\mathscr{U}\mathscr{H}}, Pr_{\mathscr{U}\mathscr{H}}, \pi_{\mathscr{U}\mathscr{H}})$ is a proof calculus.

Thus, since the map of entailment systems from many-sorted equational logic to unsorted Horn logic with equality is conservative by Corollary 28, we can apply Lemma 34 to build the proof calculus $\mathscr{PC}_{\mathscr{U}\mathscr{S}} = (\mathbf{Sign}_{\mathscr{U}\mathscr{S}}, Sen_{\mathscr{U}\mathscr{S}}, \vdash, P_{\mathscr{U}\mathscr{S}}, Pr_{\mathscr{U}\mathscr{S}}, \pi_{\mathscr{U}\mathscr{S}})$, where $P_{\mathscr{U}\mathscr{S}}$ sends any many-sorted theory T to the set of pairs consisting of a sentence $\phi \in Sen(T)$ and a proof tree t for $\Phi(T)$ s.t. the root of t is $\alpha(\phi)$, $Pr_{\mathscr{U}\mathscr{S}}$ is the identity and $\pi_{\mathscr{U}\mathscr{S}}$ is the first projection.

Note that the proof trees in $P_{\mathscr{U}\mathscr{S}}(T)$ may be (and in general are) labeled on unsorted sentences which are not the image of any many-sorted sentence.

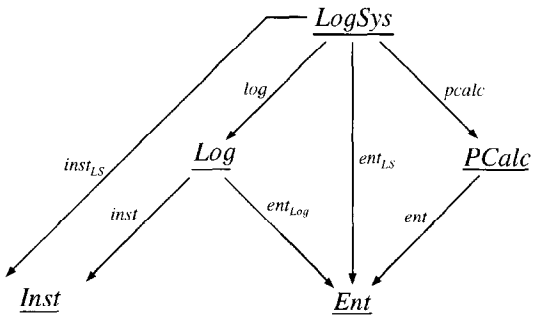
Example 36. Given a proof calculus for unsorted conditional equational logic $\mathscr{CE}$ – which we could either preferably define directly, or otherwise borrow using Lemma 34 from the one defined in the previous example for unsorted Horn logic with equality $\mathscr{L}_{\mathscr{U}\mathscr{H}}$ using the conservative embedding $\mathscr{CE} \hookrightarrow \mathscr{L}_{\mathscr{U}\mathscr{H}}$ of unsorted conditional equational logic into unsorted Horn logic with equality – then we can use the conservative map of entailments systems $\mathscr{LL} \to \mathscr{CE}$ and Lemma 34 to endow linear logic with an equational-styled proof calculus quite different from the usual sequent calculus presentation of linear logic. This new proof calculus makes available to linear logic the powerful theorem-proving techniques of the equational world.

Remark 37. Note that, since maps of proof calculi do not take into account the internal structure of the proofs, any two proof calculi having the same underlying entailment systems and the same functors proofs and natural transformation π are isomorphic. Therefore the factorization of proofs by $P = proofs$ and $Pr = Id_{\mathbf{Set}}$, made in Proposition 33, is arbitrary. In particular this factorization forgets the proof structure and hence is not always the best possible choice, but it is the only canonical choice that can be made in the general case. Therefore, the borrowing construction is not as powerful as could be desired, since only the proofs, and not really their algebraic structure, are borrowed. We conjecture that the borrowing of the algebraic structure could be accomplished by using a stronger notion of map of proof calculi in which the maps also relate explicitly the categories of proof structures.

4.4. Endowing any logical component with any missing logical structure

We can use the borrowing results obtained in Sections 4.1–4.3 to show that, indeed, all the forgetful functors between categories of logical components give rise to borrowing constructions.

Theorem 38. *All functors in the diagram*



together with their corresponding right adjoint are cartesian reflections, and therefore all the above functors are fibrations, so that any missing logical structure can be borrowed by cartesian lifting.

Proof. We have already shown in Propositions 25, 30 and 33 that *inst*, ent_{Log} and *ent* with their corresponding right adjoint are cartesian reflections. The rest of the theorem follows from the following simple observations:

- The category $\underline{LogSys}$ is the pullback $\underline{Log} \underset{\underline{Ent}}{\times} \underline{PCalc}$ in **Cat**, so that the square in the diagram is a pullback square.
- *log*, being the pullback of *ent*, is then a cartesian reflection by Proposition 24.
- Similarly, *pcalc*, being the pullback of ent_{Log}, is also a cartesian reflection for the same reason.
- All composition functors in the diagram are then cartesian reflections also, thanks to Proposition 23. □

Note that the detailed general constructions in Propositions 23 and 24 specialize in this setting to detailed borrowing constructions for all the remaining cases not considered in the previous sections. In particular, note for example that all we have already said in Sections 4.1 and 4.3 about the example relating many-sorted conditional equational logic and the unsorted Horn-clause logic with equality $\mathscr{L}_{\mathscr{U},\mathscr{H}}$ also shows that we can borrow the logical system endowing $\mathscr{L}_{\mathscr{U},\mathscr{H}}$ with proof trees to make the institution of many-sorted conditional equational logic into a logical system. Similarly, in the example relating linear logic and unsorted conditional equational logic we can now borrow a proof calculus from conditional equational logic to endow linear logic with an equational-styled logical system.

5. Concluding remarks

We have proved several general results about transportation of structure across comma categories, have specialized those results to the case of cartesian reflections, and have shown how these constructions yield a general technique for borrowing logical components from one logic for use in another logic when a map between some basic component of both exists. The universal property of the cartesian lifting that performs the borrowing in each case shows that the constructions are both natural and optimal relative to any other such constructions that could have been defined in a more ad hoc way. In addition, we have shown that the constructions are particularly well behaved in that – either in general or under natural restrictions – they preserve or yield good logical properties, such as completeness, conservativity of the extension map, or exclusive dependence of the new structure on the one being borrowed. We have also illustrated the constructions with appropriate examples.

The results in this paper provide general formal methods to achieve a greater degree of reusability across logical systems and their associated tools. They can be regarded as a step towards the goal of increasing the interoperability of formal systems, a goal of practical importance since there is frequent need in practice of rigorously relating the quite different formal specifications used to formalize complex systems at different levels of abstraction, and of correctly interoperating the tools available for each formalism. In particular, the borrowing method can be useful in the context of logical frameworks, since it supports very simple ways of representing logics into a framework and of then extending those representations to the remaining logical structure.

Given the generality of the categorical techniques used, they could in principle be used not only for the notions of mapping between logics and logical components that we have considered, but also for other axiomatic notions of mapping and logical component proposed in the literature by other authors. Exploring the applicability of our techniques to those notions seems a worthwhile research direction.

Another topic that deserves more research is studying what limits and colimits exist in categories of logics or in categories of logical components. We have made use of some limited results of a positive nature in this paper and are aware of some negative results as well. A systematic study would be very useful.

Acknowledgements

We are indebted to Narciso Martí-Oliet for his careful reading of the manuscript and for his many very helpful suggestions to improve the exposition. We cordially thank José Luiz Fiadeiro, Bart Jacobs, Eugenio Moggi, Pino Rosolini and Carolyn Talcott for their valuable suggestions that have also helped us improve the paper. We finally thank the anonymous referees for their positive suggestions and criticism.

References

[1] E. Astesiano and M. Cerioli, Relationships between logical frames, in: M. Bidoit and C. Choppy, eds., *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, Vol. 655 (Springer, Berlin, 1993) 126–143.

[2] R.M. Burstall and J.A. Goguen, The semantics of Clear, a specification language, in: D. Bjørner, ed., *Proc. 1979 Copenhagen Winter School on Abstract Software Specification*, Lecture Notes in Computer Science, Vol. 86 (Springer, Berlin, 1980) 292–332.

[3] C. Chevalley, Categories and schemes, unpublished seminar notes, Univ. of California, Berkeley, 1962.

[4] J.R.B. Cockett and R.A.G. Seely, Weakly distributive categories, in: M. Fourman, P. Johnstone and A. Pitts, eds., *Proc. Symp. on Applications of Categories in Computer Science* (Cambridge Univ. Press, Cambridge, 1992) 45–65.

[5] H. Ehrig, M. Baldamus and F. Cornelius, Theory of algebraic module specification including behavioural semantics, constraints and aspects of generalized morphisms, in: *Proc. 2nd Internat. Conf. on Algebraic Methodology and Software Technology*, Iowa City, Iowa (1991) 101–125.

[6] H. Ehrig and B. Mahr, *Fundamentals of Algebraic Specifications 1: Equations and Initial Semantics*, EATCS Monographs on Theoretical Computer Science, Vol. 6 (Springer, Berlin, 1985).

[7] J. Fiadeiro and A. Sernadas, Structuring theories on consequence, in: D. Sannella and A. Tarlecki, eds., *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, Vol. 332 (Springer, Berlin, 1987) 44–72.

[8] K. Futatsugi, J.A. Goguen, J.-P. Jouannaud and J. Meseguer, Principles of OBJ2, in: B. Reid, ed., *Proc. 12th ACM Symp. on Principles of Programming Languages* (1985) 52–66.

[9] J.Y. Girard, Linear logic, *Theoret. Comput. Sci.* **50** (1987) 1–102.

[10] J.Y. Girard, Geometry of interaction III: the general case, in: *Proc. Workshop on Linear Logic* (MIT Press, Cambridge, MA, 1994).

[11] J.A. Goguen and R.M. Burstall, Introducing institutions, in: E. Clarke and D. Kozen, eds., *Logics of Programming Workshop*, Lecture Notes in Computer Science, Vol. 164 (Springer, Berlin, 1984) 221–255.

[12] J.A. Goguen and R.M. Burstall, A study in the foundations of programming methodology: specifications, institutions, charters and parchments, in: D. Pitt, S. Abramsky, A. Poigné and D. Rydehard, eds., *Proc. Summer Workshop on Category Theory and Computer Programming*, Lecture Notes in Computer Science, Vol. 240 (Springer, Berlin, 1986) 313–333.

[13] J.A. Goguen and R.M. Burstall, Institutions: abstract model theory for specification and programming, *J. ACM* **39** (1992) 95–146.

[14] J.A. Goguen and J. Meseguer, Eqlog: equality, types, and generic modules for logic programming, in: D. DeGroot and G. Lindstrom, eds., *Logic Programming: Functions, Relations and Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1986) 295–363.

[15] J.A. Goguen, T. Winkler, J. Meseguer, K. Futatsugi and J.-P. Jouannaud, Introducing OBJ, Tech. Report SRI-CSL-92-03, SRI International, Computer Science Laboratory, 1992; to appear in J.A. Goguen, ed., *Applications of Algebraic Specification Using OBJ* (Cambridge Univ. Press, Cambridge).

[16] J.W. Gray, Fibred and cofibred categories, in: S. Eilenberg, D.K. Harrison, S. MacLane and H. Röhrl, eds., *Proc. Conf. on Categorical Algebra*, La Jolla, 1965 (Springer, Berlin, 1966) 21–83.

[17] R. Harper, D. Sannella and A. Tarlecki, Logic representation in LF, in: D.H. Pitt, ed., *Category Theory and Computer Science* Lecture Notes in Computer Science, Vol. 389 (Springer, Berlin, 1989) 250–272.

[18] H. Herrlich and G.E. Strecker, *Category Theory, An Introduction* (Heldermann, Berlin, 1979).

[19] B. Jacobs, *Categorical Logic and Type Theory* (North-Holland, Amsterdam, to be published).

[20] S. MacLane, *Categories for the Working Mathematician* (Springer, Berlin, 1971).

[21] N. Martí-Oliet and J. Meseguer, An algebraic axiomatization of linear logic models, in: G.M. Reed, A.W. Roscoe and R. Wachter, eds., *Topology and Category Theory in Computer Science* (Oxford Univ. Press, Oxford, 1991) 335–355.

[22] N. Martí-Oliet and J. Meseguer, From Petri nets to linear logic, *Math. Struct. Comput. Sci.* **1** (1991) 69–101.

[23] B. Mayoh, Galleries and institutions, Tech. Report DAIMI PB-191, Aarhus Univ., 1985.

[24] J. Meseguer, General logics, in: *Logic Colloquium* '87 (North-Holland, Amsterdam, 1989) 275–329.

[25] J. Meseguer, A logical theory of concurrent objects and its realization in the Maude language, in: G. Agha, P. Wegner and A. Yonezawa, eds., *Research Directions in Object-Oriented Programming* (MIT Press, Cambridge, MA, 1993) 314–390.

[26] J. Meseguer and N. Martí-Oliet, From abstract data types to logical frameworks, in: E. Astesiano, G. Reggio and A. Tarlecki, eds., *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, Vol. 906 (Springer, Berlin, 1995) 48–80.

[27] P. Padawitz and M. Wirsing, Completeness of many-sorted equational logic revisited, *Bull. EATCS* **24** (1984) 88–94.

[28] A. Poigné, Foundations are rich institutions, but institutions are poor foundations, in: H. Ehrig, H. Herrlich, H.-J. Kreowski and G. Preuß, eds., *Categorical Methods in Computer Science*, Lecture Notes in Computer Science, Vol. 393 (Springer, Berlin, 1989) 82–101.

[29] A. Salibra and G. Scollo, A soft stairway to institutions, in: M. Bidoit and C. Choppy, eds., *Recent Trends in Data Type Specification*, Lecture Notes in Computer Science, Vol. 655 (Springer, Berlin, 1992) 310–329.

[30] D. Sannella and A. Tarlecki, On observational equivalence and specifications, *J. Comput. System Sci.* **34** (1987) 150–178.

[31] D. Sannella and A. Tarlecki, Specifications in an arbitrary institution, *Inform. and Comput.* **76** (1988) 165–210.

[32] R.A.G. Seely, Linear logic, $*$-autonomous categories and cofree coalgebras, in: J.W. Gray and A. Scedrov, eds., *AMS Summer Research Conf. on Categories in Computer Science and Logic* (AMS, Providence, RI, 1989) 371–382.

[33] A Tarlecki, Free constructions in algebraic institutions, in: M.P. Chytil and V. Koubek, eds., *Proceedings of Mathematical Foundation of Computer Science '84*, Lecture Notes in Computer Science, Vol. 176 (Springer Verlag, Berlin, 1984) 526–534.

[34] A. Tarlecki, On the existence of free models in abstract algebraic institutions, *Theoretical Computer Science* **37** (1985) 269–304.